

# UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS Y  
COMPUTACIÓN



**UPLA**

**ASIGNATURA:** Base de Datos II

**ESTUDIANTE:** Espejo Quispe Luis Enrique

**DOCENTE:** Mg. Raúl Fernandez Bejarano

**CICLO:** V

**SECCIÓN:** A1

**HYO-2025**

### **Proyecto 1: Creación y distribución de archivos físicos de la base QhatuPeru**

Como parte de la expansión comercial de QhatuPeru, la empresa solicita la creación de una nueva base de datos que distribuya los datos entre un archivo primario, un archivo secundario y el archivo de registro de transacciones, asegurando mejor rendimiento y seguridad en el almacenamiento.

#### **Ejercicio práctico:**

- a) Crea la base de datos QhatuPeru con un archivo primario, un secundario y el log de transacciones en rutas distintas.
- b) Consulta los archivos físicos asociados a la base QhatuPeru.

### **SCRIPT EN T-SQL:**

```
USE master;
GO

-- Verificar si existe la BD y eliminarla
IF EXISTS (SELECT name FROM sys.databases WHERE name = 'QhatuPeru')
BEGIN
    ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
    DROP DATABASE QhatuPeru;
END
GO

-- Crear la base de datos con tres archivos
CREATE DATABASE QhatuPeru
ON PRIMARY
(
    NAME = 'QhatuPeru_Primary',
    FILENAME = 'C:\BaseDatos2025II\QhatuPeru_Primary.mdf',
    SIZE = 100MB,
    MAXSIZE = 500MB,
    FILEGROWTH = 50MB
),
FILEGROUP QhatuPeru_Secondary
(
    NAME = 'QhatuPeru_Secondary',
    FILENAME = 'C:\BaseDatos2025II\QhatuPeru_Secondary.ndf',
    SIZE = 100MB,
    MAXSIZE = 500MB,
    FILEGROWTH = 50MB
)
LOG ON
(
    NAME = 'QhatuPeru_Log',
    FILENAME = 'C:\BaseDatos2025II\QhatuPeru_Log.ldf',
    SIZE = 50MB,
    MAXSIZE = 200MB,
    FILEGROWTH = 25MB
);
GO
```

```
-- Consultar archivos físicos asociados
USE QhatuPeru;
GO

SELECT
    name AS NombreLogico,
    physical_name AS RutaFisica,
    type_desc AS TipoArchivo,
    size * 8 / 1024 AS TamañoMB,
    max_size * 8 / 1024 AS TamañoMaximoMB,
    growth * 8 / 1024 AS CrecimientoMB
FROM sys.database_files;
GO
```

	NombreLogico	RutaFisica	TipoArchivo	TamañoMB	TamañoMaximoMB	CrecimientoMB
1	QhatuPeru_Primary	C:\BaseDatos2025II\QhatuPeru_Primary.mdf	ROWS	100	500	50
2	QhatuPeru_Log	C:\BaseDatos2025II\QhatuPeru_Log.ldf	LOG	50	200	25
3	QhatuPeru_Secondary	C:\BaseDatos2025II\QhatuPeru_Secondary.ndf	ROWS	100	500	50

## JUSTIFICACIÓN TÉCNICA:

- Archivo primario (PRIMARY): Almacena objetos del sistema y datos de usuario
- Archivo secundario (SECONDARY): Distribuye datos para mejorar rendimiento I/O
- Log de transacciones: En disco separado para recuperación ante fallos
- Rutas distintas optimizan el acceso concurrente y minimizan cuellos de botella
- MAXSIZE previene saturación del disco
- FILEGROWTH controlado evita fragmentación excesiva

## BUENAS PRÁCTICAS:

- ✓ Separar archivos de datos y logs en discos diferentes
- ✓ Usar filegroups para organizar objetos por uso
- ✓ Establecer tamaños iniciales realistas según carga esperada
- ✓ Configurar crecimiento en MB (no porcentaje) para predecibilidad

## Proyecto 2: Ajuste de configuración y validación de propiedades de QhatuPeru

"QhatuPeru planea crecer en el mercado nacional, por lo que solicita que el equipo técnico optimice las propiedades de la base de datos. Debe cambiar la colación para soportar tildes y configurar el crecimiento automático del archivo de datos, evitando saturaciones inesperadas."

### Ejercicio práctico:

- Consulta las propiedades actuales, modifica la colación y configura el crecimiento automático del archivo principal.
- Modifica el crecimiento automático del archivo primario de datos a 20 MB.

### SCRIPT EN T-SQL:

```
USE master;
GO

-- a) Consultar propiedades actuales
SELECT
    name AS NombreBaseDatos,
    collation_name AS Colacion,
    recovery_model_desc AS ModeloRecuperacion,
    compatibility_level AS NivelCompatibilidad
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

-- Modificar la colación (requiere BD sin conexiones activas)
ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO

ALTER DATABASE QhatuPeru
COLLATE Modern_Spanish_CI_AS;
GO

ALTER DATABASE QhatuPeru SET MULTI_USER;
GO
```

0 % ▾

Results Messages

NombreBaseDatos	Colacion	ModeloRecuperacion	NivelCompatibilidad
QhatuPeru	Modern_Spanish_CI_AS	FULL	160

```
-- b) Modificar crecimiento automático del archivo principal a 20 MB
ALTER DATABASE QhatuPeru
MODIFY FILE
(
    NAME = 'QhatuPeru_Primary',
    FILEGROWTH = 20MB
);
GO

110 % < Messages
Commands completed successfully.

Completion time: 2025-11-06T10:02:49.4922719-05:00

-- Verificar cambios
USE QhatuPeru;
GO

SELECT
    name AS NombreArchivo,
    growth * 8 / 1024 AS CrecimientoMB,
    is_percent_growth AS EsPorcentaje
FROM sys.database_files
WHERE name = 'QhatuPeru_Primary';
GO

10 % < Results < Messages


|   | NombreArchivo     | CrecimientoMB | EsPorcentaje |
|---|-------------------|---------------|--------------|
| 1 | QhatuPeru_Primary | 20            | 0            |


```

## JUSTIFICACIÓN TÉCNICA :

- Modern\_Spanish\_CI\_AS: Collation que soporta tildes y ñ del español
  - CI = Case Insensitive (no distingue mayúsculas/minúsculas)
  - AS = Accent Sensitive (distingue tildes)
- Crecimiento de 20MB: Balance entre evitar fragmentación y no reservar espacio excesivo
- SINGLE\_USER temporal: Necesario para cambios estructurales críticos

## BUENAS PRÁCTICAS:

- ✓ Elegir collation apropiada desde el inicio (cambiarla después es complejo)

- ✓ Usar crecimiento en MB fijo para transacciones predecibles
- ✓ Monitorear espacio disponible con sys.database\_files
- ✓ Documentar cambios de configuración para auditorías

#### **Proyecto 3: Definición de modelo de recuperación y respaldo para QhatuPeru**

"En el área de operaciones de QhatuPeru se identificaron diferentes necesidades de recuperación de datos según el proceso. Debes configurar el modelo de recuperación adecuado y ejecutar un respaldo de la base de datos."

#### **Ejercicio práctico:**

- a) Cambia el modelo de recuperación de QhatuPeru a Simple y luego a Bulk-Logged, explicando la diferencia práctica entre ambos modelos.
- b) Realiza un respaldo completo después de cambiar al modelo FULL.

#### **SCRIPT EN T-SQL:**

#### **MODELO DE RECUPERACIÓN SIMPLE**

```

USE master;
GO

-- a) Cambiar modelo de recuperación de Simple a Bulk-Logged
ALTER DATABASE QhatuPeru
SET RECOVERY SIMPLE;
GO

-- Verificar modelo actual
SELECT
    name AS BaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

```

110 %

	BaseDatos	ModeloRecuperacion
1	QhatuPeru	SIMPLE

## MODELO DE RECUPERACIÓN BULK\_LOGGED

```
-- Verificar modelo actual
SELECT
    name AS BaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

-- Cambiar a BULK-LOGGED
ALTER DATABASE QhatuPeru
SET RECOVERY BULK_LOGGED;
GO
```

110 %

Results Messages

	BaseDatos	ModeloRecuperacion
1	QhatuPeru	BULK_LOGGED

## MODELO DE RECUPERACIÓN FULL

```
-- Verificar modelo actual
SELECT
    name AS BaseDatos,
    recovery_model_desc AS ModeloRecuperacion
FROM sys.databases
WHERE name = 'QhatuPeru';
GO

-- Cambiar a FULL para demostrar
ALTER DATABASE QhatuPeru
SET RECOVERY FULL;
GO
```

110 %

Results Messages

	BaseDatos	ModeloRecuperacion
1	QhatuPeru	FULL

## RESPALDO COMPLETO

```
-- b) Realizar respaldo completo después de cambiar a modelo FULL
BACKUP DATABASE QhatuPeru
TO DISK = 'C:\SQLBackups\QhatuPeru_Full.bak'
WITH
    FORMAT,
    NAME = 'QhatuPeru-Respaldo Completo',
    DESCRIPTION = 'Respaldo completo después de configurar modelo FULL',
    COMPRESSION,
    STATS = 10;
GO

-- Respaldo diferencial (ejemplo adicional)
BACKUP DATABASE QhatuPeru
TO DISK = 'C:\SQLBackups\QhatuPeru_Diff.bak'
WITH
    DIFFERENTIAL,
    NAME = 'QhatuPeru-Respaldo Diferencial',
    COMPRESSION,
    STATS = 10;
GO

-- Respaldo de log de transacciones
BACKUP LOG QhatuPeru
TO DISK = 'C:\SQLBackups\QhatuPeru_Log.trn'
WITH
    NAME = 'QhatuPeru-Respaldo de Log',
    COMPRESSION,
    STATS = 10;
GO
11 % < > Messages
27 percent processed.
52 percent processed.
69 percent processed.
71 percent processed.
83 percent processed.
93 percent processed.
100 percent processed.
Processed 456 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 24 pages for database 'QhatuPeru', file 'QhatuPeru_Secondary' on file 1.
Processed 1 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP DATABASE successfully processed 481 pages in 0.024 seconds (156.811 MB/sec).
32 percent processed.
43 percent processed.
54 percent processed.
65 percent processed.
87 percent processed.
98 percent processed.
100 percent processed.
Processed 80 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 16 pages for database 'QhatuPeru', file 'QhatuPeru_Secondary' on file 1.
Processed 1 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 97 pages in 0.013 seconds (57.804 MB/sec).
100 percent processed.
Processed 3 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP LOG successfully processed 3 pages in 0.003 seconds (6.347 MB/sec).
```

## JUSTIFICACIÓN TÉCNICA:

- Modelo FULL: Recomendado para QhatuPeru (empresa comercial con datos críticos)
- BACKUP WITH COMPRESSION: Reduce tamaño de archivos de respaldo ~50-70%
- STATS = 10: Muestra progreso cada 10% (útil para backups grandes)
- Estrategia recomendada:
  - \* Respaldo FULL: Semanal (domingos)
  - \* Respaldo DIFERENCIAL: Diario (lunes a sábado)

\* Respaldo LOG: Cada 15-30 minutos en horario laboral

## BUENAS PRÁCTICAS:

- ✓ Almacenar backups en ubicación diferente al servidor
- ✓ Validar backups con RESTORE VERIFYONLY
- ✓ Automatizar respaldos con SQL Server Agent
- ✓ Mantener política de retención (ej: 30 días)
- ✓ Documentar RTO (Recovery Time Objective) y RPO (Recovery Point Objective)

### Proyecto 4: Implementación de roles y usuarios para seguridad en QhatuPeru

"Con el crecimiento de los equipos de ventas y atención al cliente, QhatuPeru solicita la creación de usuarios con roles diferenciados: cajeros acceden solo a consulta, administradores tienen control total y el gerente requiere acceso a reportes."

#### Ejercicio práctico:

- a) Crea el usuario VendedorQhatu y asígnalo al rol db\_datawriter para registrar ventas.
- b) Crea el usuario ConsultaCliente y asígnalo solo al rol db\_datareader.

## SCRIPT EN T-SQL:

### CREACIÓN DE ROLES

The screenshot shows a SQL Server Management Studio (SSMS) interface. The main pane displays the following T-SQL script:

```
-- 208 USE QhatuPeru;
-- 209 GO
-- 210
-- 211 -- Crear roles de base de datos
-- 212 CREATE ROLE db_datawriter AUTHORIZATION dbo;
-- 213 CREATE ROLE db_datareader AUTHORIZATION dbo;
-- 214 GO
-- 215
```

The status bar at the bottom left indicates "91 %". Below the main pane is a "Messages" window with the following output:

Commands completed successfully.  
Completion time: 2025-11-06T12:33:36.0196890-05:00

## ROL VENDEDOR

```
215 -- a) Crear usuario VendedorQhatu con rol de escritura
216 CREATE LOGIN VendedorQhatu WITH PASSWORD = 'Vendedor#2024$';
217 GO
218
219 USE QhatuPeru;
220 GO
221
222 CREATE USER VendedorQhatu FOR LOGIN VendedorQhatu;
223 GO
224
225 -- Asignar rol db_datawriter (permite INSERT, UPDATE, DELETE)
226 ALTER ROLE db_datawriter ADD MEMBER VendedorQhatu;
227 [ALTER ROLE db_datareader ADD MEMBER VendedorQhatu;
228 GO
229
230 PRINT 'Usuario VendedorQhatu creado con permisos de lectura y escritura';
231 GO
232
```

1 % ▾

Messages

Usuario VendedorQhatu creado con permisos de lectura y escritura

Completion time: 2025-11-06T12:35:01.6635284-05:00

## ROL CLIENTE

```
233 -- b) Crear usuario ConsultaCliente con rol de solo lectura
234 CREATE LOGIN ConsultaCliente WITH PASSWORD = 'Consulta#2024$';
235 GO
236
237 USE QhatuPeru;
238 GO
239
240 CREATE USER ConsultaCliente FOR LOGIN ConsultaCliente;
241 GO
242
243 -- Asignar rol db_datareader (solo SELECT)
244 ALTER ROLE db_datareader ADD MEMBER ConsultaCliente;
245 GO
246
247 PRINT 'Usuario ConsultaCliente creado con permisos de solo lectura';
248 GO
249
250
```

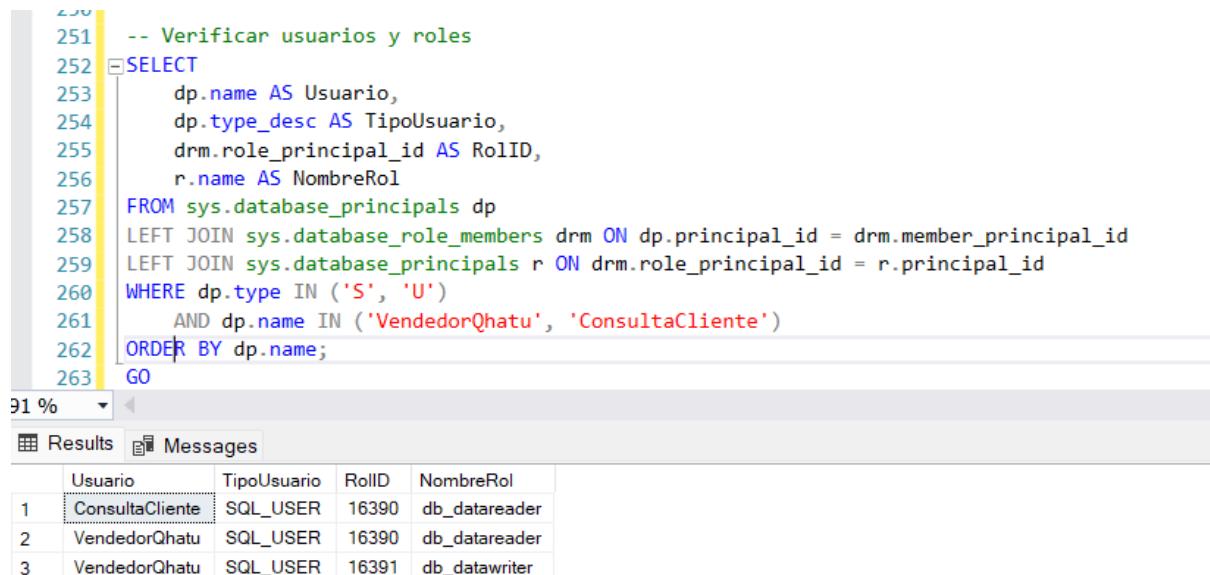
1 % ▾

Messages

Usuario ConsultaCliente creado con permisos de solo lectura

Completion time: 2025-11-06T12:35:46.2693938-05:00

## VERIFICACIÓN



```
251  -- Verificar usuarios y roles
252  SELECT
253      dp.name AS Usuario,
254      dp.type_desc AS TipoUsuario,
255      drm.role_principal_id AS RolID,
256      r.name AS NombreRol
257  FROM sys.database_principals dp
258  LEFT JOIN sys.database_role_members drm ON dp.principal_id = drm.member_principal_id
259  LEFT JOIN sys.database_principals r ON drm.role_principal_id = r.principal_id
260  WHERE dp.type IN ('S', 'U')
261      AND dp.name IN ('VendedorQhatu', 'ConsultaCliente')
262  ORDER BY dp.name;
263  GO
```

Results

	Usuario	TipoUsuario	RolID	NombreRol
1	ConsultaCliente	SQL_USER	16390	db_datareader
2	VendedorQhatu	SQL_USER	16390	db_datareader
3	VendedorQhatu	SQL_USER	16391	db_datawriter

## JUSTIFICACIÓN TÉCNICA:

- db\_datawriter: Permite INSERT, UPDATE, DELETE en todas las tablas
- db\_datareader: Permite SELECT en todas las tablas
- Separación de responsabilidades según rol laboral
- VendedorQhatu: Necesita registrar ventas (escritura)
- ConsultaCliente: Solo consulta información (lectura)
- Contraseñas complejas cumplen políticas de seguridad

## BUENAS PRÁCTICAS:

- ✓ Aplicar principio de mínimo privilegio
- ✓ No usar cuenta 'sa' para operaciones cotidianas
- ✓ Implementar roles personalizados según procesos de negocio
- ✓ Auditar accesos con SQL Server Audit
- ✓ Rotar contraseñas periódicamente
- ✓ Usar autenticación de Windows cuando sea posible



### Proyecto 5: Configuración granular de permisos en el módulo de ventas de QhatuPeru

"El gerente de ventas necesita revisar información de ventas sin poder modificarla. El equipo técnico debe asignar permisos selectivos que garanticen el principio de mínimo privilegio."

#### Ejercicio práctico:

- Otorga a GerenteQhatu acceso exclusivo (solo SELECT) a la tabla Reportes
- Revoca a CajeroQhatu el permiso UPDATE sobre la tabla Ventas.

#### SCRIPT EN T-SQL:

#### CREACIÓN DE TABALAS VENTAS Y REPORTES (NO INCLUIDOS EN LAS TABLAS DE LA BASE DE DATOS QHATUPERU)

The screenshot shows a SQL script being run in SQL Server Management Studio. The script creates two tables, Ventas and Reportes, in the QhatuPeru database. The 'Messages' pane at the bottom indicates that the commands completed successfully.

```
USE QhatuPeru;
GO

IF OBJECT_ID('Ventas', 'U') IS NULL
BEGIN
    CREATE TABLE Ventas (
        VentaID INT PRIMARY KEY IDENTITY(1,1),
        ClienteID INT NOT NULL,
        FechaVenta DATETIME DEFAULT GETDATE(),
        MontoTotal DECIMAL(10,2) NOT NULL,
        EstadoVenta VARCHAR(20) DEFAULT 'Pendiente'
    );
END
GO

IF OBJECT_ID('Reportes', 'U') IS NULL
BEGIN
    CREATE TABLE Reportes (
        ReporteID INT PRIMARY KEY IDENTITY(1,1),
        TipoReporte VARCHAR(50),
        FechaGeneracion DATETIME DEFAULT GETDATE(),
        Contenido NVARCHAR(MAX)
    );
END
GO
```

83 % ▾

Messages

Commands completed successfully.

Completion time: 2025-11-06T10:25:05.6034665-05:00

#### CREACIÓN DE USUARIOS

```
-- Crear usuarios para el proyecto
IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = 'GerenteQhatu')
BEGIN
    CREATE LOGIN GerenteQhatu WITH PASSWORD = 'Gerente#2024$';
    CREATE USER GerenteQhatu FOR LOGIN GerenteQhatu;
END
GO

IF NOT EXISTS (SELECT * FROM sys.database_principals WHERE name = 'CajeroQhatu')
BEGIN
    CREATE LOGIN CajeroQhatu WITH PASSWORD = 'Cajero#2024$';
    CREATE USER CajeroQhatu FOR LOGIN CajeroQhatu;
END
GO
```

3 % ▶ Messages  
Commands completed successfully.  
Completion time: 2025-11-06T10:25:05.6034665-05:00

## PERMISOS OTORGADOS

```
-- a) Otorgar a GerenteQhatu acceso exclusivo (solo SELECT) a tabla Reportes
GRANT SELECT ON Reportes TO GerenteQhatu;
GO

-- Denegar otros permisos explícitamente
DENY INSERT, UPDATE, DELETE ON Reportes TO GerenteQhatu;
GO

PRINT 'GerenteQhatu tiene acceso exclusivo de SELECT a tabla Reportes';
GO
```

91 % ▶ Messages  
GerenteQhatu tiene acceso exclusivo de SELECT a tabla Reportes  
Completion time: 2025-11-06T10:27:22.0225716-05:00

```

-- b) Revocar permiso UPDATE sobre tabla Ventas a CajeroQhatu
-- Primero otorgar permisos básicos al cajero
GRANT SELECT, INSERT ON Ventas TO CajeroQhatu;
GO

-- Ahora revocar UPDATE específicamente
REVOKE UPDATE ON Ventas FROM CajeroQhatu;
GO

PRINT 'CajeroQhatu puede consultar e insertar ventas, pero no modificarlas';
GO

```

1 %

Messages

CajeroQhatu puede consultar e insertar ventas, pero no modificarlas

Completion time: 2025-11-06T10:28:27.9489763-05:00

## VERIFICACIÓN

```

-- Verificar permisos otorgados
SELECT
    USER_NAME(grantee_principal_id) AS Usuario,
    OBJECT_NAME(major_id) AS Objeto,
    permission_name AS Permiso,
    state_desc AS Estado
FROM sys.database_permissions
WHERE USER_NAME(grantee_principal_id) IN ('GerenteQhatu', 'CajeroQhatu')
    AND class_desc = 'OBJECT_OR_COLUMN'
ORDER BY Usuario, Objeto, Permiso;
GO

```

91 %

Results

	Usuario	Objeto	Permiso	Estado
1	CajeroQhatu	Ventas	INSERT	GRANT
2	CajeroQhatu	Ventas	SELECT	GRANT
3	GerenteQhatu	Reportes	DELETE	DENY
4	GerenteQhatu	Reportes	INSERT	DENY
5	GerenteQhatu	Reportes	SELECT	GRANT
6	GerenteQhatu	Reportes	UPDATE	DENY

## **JUSTIFICACIÓN TÉCNICA:**

- GRANT: Otorga permisos específicos
- DENY: Deniega explícitamente (prevalece sobre GRANT)
- REVOKE: Quita permisos previamente otorgados
- Permisos granulares por tabla controlan acceso preciso
- GerenteQhatu: Solo lectura de reportes (no puede alterarlos)
- CajeroQhatu: Registra ventas pero no puede modificar historial

## **DIFERENCIAS CLAVE:**

- REVOKE vs DENY:
  - \* REVOKE: Elimina permiso, usuario puede heredar de rol
  - \* DENY: Bloquea explícitamente, prevalece sobre herencia

## **BUENAS PRÁCTICAS:**

- ✓ Documentar matriz de permisos por rol
- ✓ Preferir REVOKE sobre DENY (menos confuso en herencia)
- ✓ Usar roles en lugar de permisos directos a usuarios
- ✓ Implementar permisos a nivel columna para datos sensibles
- ✓ Revisar permisos con sp\_helpprotect regularmente

### **Proyecto 6: Identificación y solución de procesos lentos en QhatuPeru**

"Durante una campaña promocional, el sistema presenta lentitud. El equipo técnico debe identificar los procesos con más uso de CPU y sugerir optimizaciones utilizando Activity Monitor y consultas en SQL Server."

#### **Ejercicio práctico:**

- a) Usa Activity Monitor para identificar tres procesos que consuman más CPU durante una operación masiva.
- b) Consulta desde T-SQL los bloqueos actuales en la base de datos.

## SCRIPT EN T-SQL:

```
-- a) Script para identificar tres procesos que consumen más CPU
SELECT TOP 3
    s.session_id AS SessionID,
    s.login_name AS Usuario,
    s.host_name AS Host,
    r.status AS Estado,
    r.command AS Comando,
    r.cpu_time AS TiempoCPU_ms,
    r.total_elapsed_time AS TiempoTotal_ms,
    t.text AS ConsultaSQL,
    DB_NAME(r.database_id) AS BaseDatos,
    r.blocking_session_id AS BloqueadoPor
FROM sys.dm_exec_sessions s
INNER JOIN sys.dm_exec_requests r ON s.session_id = r.session_id
CROSS APPLY sys.dm_exec_sql_text(r.sql_handle) t
WHERE s.is_user_process = 1
ORDER BY r.cpu_time DESC;
GO
```

SessionID	Usuario	Host	Estado	Comando	TiempoCPU_ms	TiempoTotal_ms	ConsultaSQL	BaseDatos	BloqueadoPor
68	LAB04-PC16\USER 17	LAB04-PC16	running	SELECT	2	2	SELECT TOP 3 s.session_id AS SessionID, ...	QhatuPeru	0

```
-- b) Consultar bloqueos actuales en la base de datos
SELECT
    blocked.session_id AS SesionBloqueada,
    blocked_sql.text AS ConsultaBloqueada,
    blocker.session_id AS SesionBloqueadora,
    blocker_sql.text AS ConsultaBloqueadora,
    blocked.wait_type AS TipoEspera,
    blocked.wait_time AS TiempoEspera_ms,
    blocked.wait_resource AS RecursoBloqueado
FROM sys.dm_exec_requests blocked
CROSS APPLY sys.dm_exec_sql_text(blocked.sql_handle) blocked_sql
INNER JOIN sys.dm_exec_requests blocker ON blocked.blocking_session_id = blocker.session_id
CROSS APPLY sys.dm_exec_sql_text(blocker.sql_handle) blocker_sql
WHERE blocked.blocking_session_id <> 0;
GO
```

SesionBloqueada	ConsultaBloqueada	SesionBloqueadora	ConsultaBloqueadora	TipoEspera	TiempoEspera_ms	RecursoBloqueado

## JUSTIFICACIÓN TÉCNICA:

- Activity Monitor: Herramienta gráfica de SSMS para monitoreo en tiempo real
- sys.dm\_exec\_requests: DMV que muestra solicitudes activas actualmente
- sys.dm\_exec\_query\_stats: Estadísticas agregadas de consultas ejecutadas
- Métricas clave:
  - cpu\_time: Tiempo de CPU consumido
  - total\_elapsed\_time: Tiempo total transcurrido
  - blocking\_session\_id: Identifica bloqueos

## OPTIMIZACIONES SUGERIDAS:

1. Para consultas lentas con alto CPU:

- a. Crear índices apropiados
  - b. Reescribir consulta para evitar scans completos
  - c. Actualizar estadísticas (UPDATE STATISTICS)
2. Para bloqueos:
- a. Reducir tiempo de transacciones
  - b. Usar nivel de aislamiento apropiado (READ COMMITTED SNAPSHOT)
  - c. Implementar índices para reducir bloqueos

## **BUENAS PRÁCTICAS:**

- ✓ Monitorear DMVs regularmente con SQL Server Agent
- ✓ Configurar alertas para uso de CPU > 80%
- ✓ Implementar Query Store para análisis histórico
- ✓ Usar Extended Events para capturar eventos problemáticos
- ✓ Establecer umbrales de timeout para consultas

### **Proyecto 7: Automatización de respaldos y limpieza del sistema QhatuPeru**

"Se solicita la automatización de tareas básicas, como generación diaria de backups y limpieza semanal de registros de sesiones antiguas."

#### **Ejercicio práctico:**

- a) Crea un Job en SQL Server Agent que realice un respaldo diario automático de QhatuPeru.  
(El paso se hace en la interfaz gráfica, pero el script es:)
- b) Diseña y programa un Job que elimine registros de la tabla Sesiones con más de 15 días de antigüedad cada semana.

## **SCRIPT EN T-SQL:**

### **CREACIÓN DE LA TABLA SESIONES**

```
USE QhatuPeru;
GO

-- Crear tabla Sesiones para el ejercicio
IF OBJECT_ID('Sesiones', 'U') IS NULL
BEGIN
    CREATE TABLE Sesiones (
        SesionID INT PRIMARY KEY IDENTITY(1,1),
        UsuarioID INT NOT NULL,
        FechaInicio DATETIME DEFAULT GETDATE(),
        FechaFin DATETIME NULL,
        DireccionIP VARCHAR(45),
        Estado VARCHAR(20) DEFAULT 'Activa'
    );
    -- Insertar datos de prueba
    INSERT INTO Sesiones (UsuarioID, FechaInicio, FechaFin, DireccionIP, Estado)
    VALUES
        (1, DATEADD(DAY, -20, GETDATE()), DATEADD(DAY, -20, GETDATE()), '192.168.1.10', 'Cerrada'),
        (2, DATEADD(DAY, -10, GETDATE()), DATEADD(DAY, -10, GETDATE()), '192.168.1.11', 'Cerrada'),
        (3, DATEADD(DAY, -5, GETDATE()), NULL, '192.168.1.12', 'Activa');
END
GO
```

91 %

Messages

```
(3 rows affected)
Completion time: 2025-11-06T10:37:38.2884949-05:00
```

## CREACIÓN DEL JOB DE RESPLADO

```

-- a) Crear Job para respaldo diario automático de QhatuPeru
USE msdb;
GO

-- Eliminar job si existe
IF EXISTS (SELECT * FROM msdb.dbo.sysjobs WHERE name = 'Respaldo_Diario_QhatuPeru')
BEGIN
    EXEC msdb.dbo.sp_delete_job @job_name = 'Respaldo_Diario_QhatuPeru';
END
GO

-- Crear el Job
EXEC msdb.dbo.sp_add_job
    @job_name = 'Respaldo_Diario_QhatuPeru',
    @enabled = 1,
    @description = 'Respaldo completo diario de la base de datos QhatuPeru',
    @category_name = 'Database Maintenance';
GO

-- Agregar paso al Job (el script de backup)
EXEC msdb.dbo.sp_add_jobstep
    @job_name = 'Respaldo_Diario_QhatuPeru',
    @step_name = 'Ejecutar_Backup_Completo',
    @subsystem = 'TSQL',
    @command =
        '
DECLARE @BackupPath NVARCHAR(500);
DECLARE @fileName NVARCHAR(500);

SET @BackupPath = ''C:\SQLBackups\'';
SET @fileName = @BackupPath + ''QhatuPeru_'' +
    CONVERT(VARCHAR, GETDATE(), 112) + ''_'' +
    REPLACE(CONVERT(VARCHAR, GETDATE(), 108), '':'', '') + '' .bak'';

BACKUP DATABASE QhatuPeru
TO DISK = @fileName
WITH
    FORMAT,
    COMPRESSION,
    NAME = ''QhatuPeru-Respaldo Diario'',
    DESCRIPTION = ''Respaldo automático diario'',
    STATS = 10;

PRINT ''Respaldo completado: '' + @fileName;
',
    @retry_attempts = 3,
    @retry_interval = 5;
GO

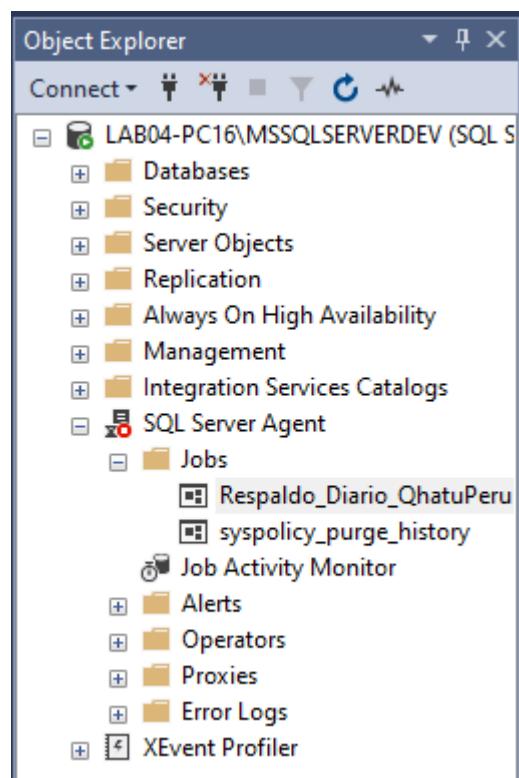
```

75 %

Messages

Commands completed successfully.

Completion time: 2025-11-06T10:40:41.6666591-05:00



```
-- Crear schedule (programación diaria a las 2:00 AM)
EXEC msdb.dbo.sp_add_schedule
    @schedule_name = 'Diario_2AM',
    @freq_type = 4, -- Diario
    @freq_interval = 1,
    @active_start_time = 020000; -- 02:00:00 AM
GO

-- Asociar schedule al job
EXEC msdb.dbo.sp_attach_schedule
    @job_name = 'Respaldo_Diario_QhatuPeru',
    @schedule_name = 'Diario_2AM';
GO

-- Asignar job al servidor local
EXEC msdb.dbo.sp_add_jobserver
    @job_name = 'Respaldo_Diario_QhatuPeru',
    @server_name = '(local)';
GO

PRINT 'Job de respaldo diario creado exitosamente';
GO
```

10 %

Messages

SQLServerAgent is not currently running so it cannot be notified of this action.

Job de respaldo diario creado exitosamente

Completion time: 2025-11-06T10:43:19.8695834-05:00

## CREACION DEL JOB DE LIMPIEZA

```

-- b) Job para eliminar registros antiguos de tabla Sesiones (semanal)

-- Eliminar job si existe
IF EXISTS (SELECT * FROM msdb.dbo.sysjobs WHERE name = 'Limpieza_Sesiones_Antiguas')
BEGIN
    EXEC msdb.dbo.sp_delete_job @job_name = 'Limpieza_Sesiones_Antiguas';
END
GO

-- Crear el Job de limpieza
EXEC msdb.dbo.sp_add_job
    @job_name = 'Limpieza_Sesiones_Antiguas',
    @enabled = 1,
    @description = 'Elimina sesiones con más de 15 días de antigüedad',
    @category_name = 'Database Maintenance';
GO

-- Agregar paso al Job
EXEC msdb.dbo.sp_add_jobstep
    @job_name = 'Limpieza_Sesiones_Antiguas',
    @step_name = 'Eliminar_Sesiones_Viejas',
    @subsystem = 'TSQL',
    @command =
        DECLARE @FechaLimite DATETIME;
        DECLARE @RegistrosEliminados INT;

        SET @FechaLimite = DATEADD(DAY, -15, GETDATE());

        DELETE FROM Sesiones
        WHERE FechaInicio < @FechaLimite
        AND Estado = 'Cerrada';

        SET @RegistrosEliminados = @@ROWCOUNT;

        PRINT 'Limpieza completada. Registros eliminados: ' +
            CAST(@RegistrosEliminados AS VARCHAR(10));

        -- Registrar en tabla de log (opcional)
        IF OBJECT_ID('LogMantenimiento', 'U') IS NOT NULL
        BEGIN
            INSERT INTO LogMantenimiento (TipoMantenimiento, FechaEjecucion, RegistrosAfectados)
                VALUES ('Limpieza Sesiones', GETDATE(), @RegistrosEliminados);
        END
    ',
    @database_name = 'QhatuPeru',
    @retry_attempts = 2,
    @retry_interval = 5;
GO

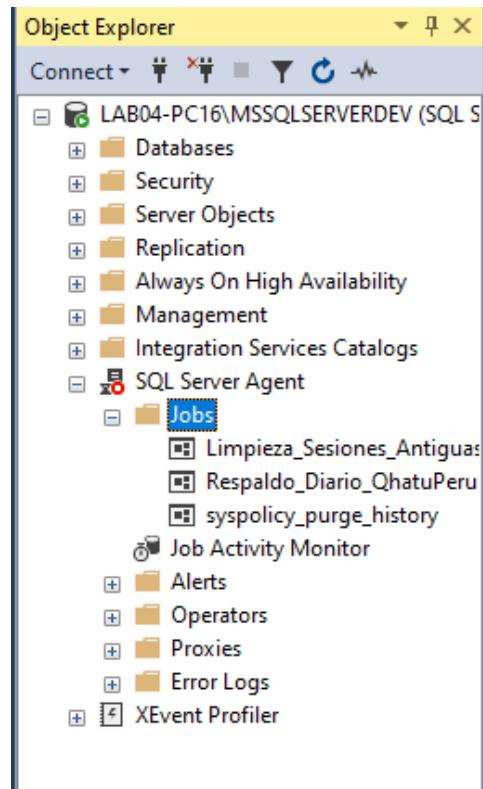
```

15 %

Messages

Commands completed successfully.

Completion time: 2023-11-06T10:44:20.1400331-05:00



```

-- Crear schedule (semanal: domingos a las 3:00 AM)
EXEC msdb.dbo.sp_add_schedule
    @schedule_name = 'Semanal_Domingo_3AM',
    @freq_type = 8, -- Semanal
    @freq_interval = 1, -- Domingo
    @freq_recurrence_factor = 1,
    @active_start_time = 030000; -- 03:00:00 AM
GO

-- Asociar schedule al job
EXEC msdb.dbo.sp_attach_schedule
    @job_name = 'Limpieza_Sesiones_Antiguas',
    @schedule_name = 'Semanal_Domingo_3AM';
GO

-- Asignar job al servidor local
EXEC msdb.dbo.sp_add_jobserver
    @job_name = 'Limpieza_Sesiones_Antiguas',
    @server_name = '(local)';
GO

PRINT 'Job de limpieza semanal creado exitosamente';
GO

```

00 % ▶

Messages

Job de limpieza semanal creado exitosamente

Completion time: 2025-11-06T10:46:47.3875110-05:00

## VERIFICACIÓN

```

-- Verificar Jobs creados
SELECT
    j.name AS NombreJob,
    j.enabled AS Habilitado,
    j.description AS Descripcion,
    s.name AS Schedule,
    CASE s.freq_type
        WHEN 4 THEN 'Diario'
        WHEN 8 THEN 'Semanal'
        WHEN 16 THEN 'Mensual'
    END AS Frecuencia
FROM msdb.dbo.sysjobs j
LEFT JOIN msdb.dbo.sysjobschedules js ON j.job_id = js.job_id
LEFT JOIN msdb.dbo.sysschedules s ON js.schedule_id = s.schedule_id
WHERE j.name IN ('Respaldo_Diario_QhatuPeru', 'Limpieza_Sesiones_Antiguas');
GO

```

100 % ▶

Results

	NombreJob	Habilitado	Descripcion	Schedule	Frecuencia
1	Respaldo_Diario_QhatuPeru	1	Respaldo completo diario de la base de datos Qha...	Diario_2AM	Diario
2	Limpieza_Sesiones_Antiguas	1	Elimina sesiones con más de 15 días de antigüedad	Semanal_Domingo_3AM	Semanal

## JUSTIFICACIÓN TÉCNICA:

- SQL Server Agent: Servicio que ejecuta tareas programadas
- Jobs: Unidades de trabajo automatizadas con pasos y schedules
- Respaldo diario: Protege datos críticos de QhatuPeru
- Limpieza semanal: Optimiza rendimiento eliminando datos históricos

## **COMPONENTES DE UN JOB:**

1. Job: Contenedor principal
2. Steps: Pasos individuales a ejecutar
3. Schedules: Programación temporal
4. Alerts: Notificaciones en caso de fallo (opcional)

## **CONFIGURACIÓN DE FRECUENCIAS:**

- @freq\_type:
  - 4 = Diario
  - 8 = Semanal
  - 16 = Mensual
- @freq\_interval: Día específico (1=Domingo, 2=Lunes, etc.)
- @active\_start\_time: Hora en formato HHMMSS (020000 = 2:00 AM)

## **BUENAS PRÁCTICAS:**

- ✓ Programar backups en horarios de baja actividad
- ✓ Implementar retry\_attempts para tolerancia a fallos
- ✓ Registrar ejecuciones en tabla de auditoría
- ✓ Configurar notificaciones por email (Database Mail)
- ✓ Verificar espacio en disco antes de backups
- ✓ Rotar archivos de backup antiguos
- ✓ Documentar strategy de retención

### Proyecto 8: Registro de estados en pedidos en QhatuPeru

"Con la finalidad de optimizar la trazabilidad de los envíos, se solicita agregar una columna 'EstadoEnvio' en la tabla Pedidos, permitiendo identificar el estado actual de cada pedido."

#### Ejercicio práctico:

- a) Agrega la columna "Prioridad" tipo INT a la tabla Pedidos.
- b) Elimina la columna "EstadoEnvio" de la tabla Pedidos.

## SCRIPT EN T-SQL:

### CREACIÓN DE LA TABLA PEDIDOS (NO INCLUIDO ORIGINALMENTE EN LA BASE DE DATOS QHATUPERU)

```
--  
611 USE QhatuPeru;  
612 GO  
613  
614 -- Crear tabla Pedidos si no existe  
615 IF OBJECT_ID('Pedidos', 'U') IS NULL  
616 BEGIN  
617 CREATE TABLE Pedidos (  
618 PedidoID INT PRIMARY KEY IDENTITY(1,1),  
619 ClienteID INT NOT NULL,  
620 FechaPedido DATETIME DEFAULT GETDATE(),  
621 MontoTotal DECIMAL(10,2) NOT NULL,  
622 EstadoEnvio VARCHAR(50) NULL,  
623 FechaEntrega DATETIME NULL,  
624 DireccionEnvio NVARCHAR(200)  
625 );  
626  
627 -- Insertar datos de prueba  
628 INSERT INTO Pedidos (ClienteID, FechaPedido, MontoTotal, EstadoEnvio)  
VALUES  
629     (1, GETDATE(), 150.50, 'En preparación'),  
630     (2, DATEADD(DAY, -1, GETDATE()), 320.00, 'Enviado'),  
631     (3, DATEADD(DAY, -3, GETDATE()), 89.99, 'Entregado');  
632  
633 END  
634 GO  
635
```

3 % ▾

Messages

(3 rows affected)

Completion time: 2025-11-06T12:59:15.2192492-05:00

```
636 -- a) Agregar columna "Prioridad" tipo INT a la tabla Pedidos
637 ALTER TABLE Pedidos
638 ADD Prioridad INT NULL;
639 GO
640
641 -- Actualizar valores por defecto según lógica de negocio
642 UPDATE Pedidos
643 SET Prioridad = CASE
644 WHEN MontoTotal >= 500 THEN 1 -- Alta prioridad
645 WHEN MontoTotal >= 200 THEN 2 -- Media prioridad
646 ELSE 3 -- Baja prioridad
647 END;
648 GO
649
```

1 % ▾

Messages

(3 rows affected)

Completion time: 2025-11-06T13:03:47.9160553-05:00

```
647
650 -- Crear constraint para validar valores de Prioridad
651 ALTER TABLE Pedidos
652 ADD CONSTRAINT CK_Pedidos_Prioridad
653 CHECK (Prioridad BETWEEN 1 AND 3);
654 GO
655
656 -- Agregar valor por defecto para nuevos registros
657 ALTER TABLE Pedidos
658 ADD CONSTRAINT DF_Pedidos_Prioridad DEFAULT 3 FOR Prioridad;
659 GO
660
661 PRINT 'Columna Prioridad agregada exitosamente';
662 GO
663
```

% ▾

Messages

Columna Prioridad agregada exitosamente

Completion time: 2025-11-06T13:04:29.9153256-05:00

## VERIFICACIÓN DE DATOS

```
664 -- b) Eliminar columna "EstadoEnvio" de la tabla Pedidos
665 -- Primero verificar datos existentes para migración
666 SELECT
667     PedidoID,
668     EstadoEnvio,
669     COUNT(*) AS Cantidad
670 FROM Pedidos
671 GROUP BY PedidoID, EstadoEnvio;
672 GO
```

91 %

Results Messages

	PedidoID	EstadoEnvio	Cantidad
1	1	En preparación	1
2	2	Enviado	1
3	3	Entregado	1

## ELIMINACIÓN DE LA COLUMNA

```
674 -- Eliminar la columna
675 ALTER TABLE Pedidos
676 DROP COLUMN EstadoEnvio;
677 GO
678
679 PRINT 'Columna EstadoEnvio eliminada exitosamente';
680 GO
```

L %

Messages

Columna EstadoEnvio eliminada exitosamente

Completion time: 2025-11-06T13:06:02.4171865-05:00

## VERIFICACIÓN DEL ESTADO

```

681
682 -- Verificar estructura final de la tabla
683 SELECT
684     COLUMN_NAME AS NombreColumna,
685     DATA_TYPE AS TipoDatos,
686     CHARACTER_MAXIMUM_LENGTH AS Longitud,
687     IS_NULLABLE AS Nullable,
688     COLUMN_DEFAULT AS ValorPorDefecto
689 FROM INFORMATION_SCHEMA.COLUMNS
690 WHERE TABLE_NAME = 'Pedidos'
691 ORDER BY ORDINAL_POSITION;
692 GO
693

```

1 %

	NombreColumna	TipoDatos	Longitud	Nullable	ValorPorDefecto
1	PedidoID	int	NULL	NO	NULL
2	ClienteID	int	NULL	NO	NULL
3	FechaPedido	datetime	NULL	YES	(getdate())
4	MontoTotal	decimal	NULL	NO	NULL
5	FechaEntrega	datetime	NULL	YES	NULL
6	DireccionEnvio	nvarchar	200	YES	NULL
7	Prioridad	int	NULL	YES	((3))

## JUSTIFICACIÓN TÉCNICA:

- Columna Prioridad: Permite optimizar procesamiento de pedidos urgentes
- Eliminación de EstadoEnvio: Evita redundancia si existe tabla histórica
- Constraint CHECK: Garantiza integridad de datos (solo valores 1, 2, 3)
- Tabla EstadosPedido: Registra trazabilidad completa de estados

## VENTAJAS DEL NUEVO DISEÑO:

1. Historial completo de cambios de estado
2. Auditoría de quién realizó cambios
3. Mejor rendimiento con índices apropiados
4. Flexibilidad para agregar nuevos estados sin alterar estructura

## BUENAS PRÁCTICAS:

- ✓ Migrar datos antes de eliminar columnas
- ✓ Documentar razones de cambios estructurales
- ✓ Usar tablas de auditoría para campos que cambian frecuentemente
- ✓ Implementar foreign keys para integridad referencial
- ✓ Crear índices en columnas de búsqueda frecuente

#### **Proyecto 9: Implementación de registros automáticos de modificaciones en QhatuPeru**

"Es necesario auditar todas las modificaciones realizadas en la tabla Clientes para cumplir con políticas internas de seguridad y protección de datos."

#### **Ejercicio práctico:**

- a) Crea una tabla AuditoriaClientes para registrar cambios en Clientes.
- b) Desarrolla un trigger que registre en AuditoriaClientes cada eliminación de registros en Clientes.

## **SCRIPT EN T-SQL:**

### **CREACIÓN DE LA TABLA CLIENTES**

```

720
721 USE QhatuPeru;
722 GO
723
724 -- Crear tabla Clientes si no existe
725 IF OBJECT_ID('Clientes', 'U') IS NULL
726 BEGIN
727 CREATE TABLE Clientes (
728     ClienteID INT PRIMARY KEY IDENTITY(1,1),
729     Nombre NVARCHAR(100) NOT NULL,
730     Apellido NVARCHAR(100) NOT NULL,
731     Email NVARCHAR(100) UNIQUE,
732     Telefono VARCHAR(20),
733     FechaRegistro DATETIME DEFAULT GETDATE(),
734     Activo BIT DEFAULT 1
735 );
736
737 -- Insertar datos de prueba
738 INSERT INTO Clientes (Nombre, Apellido, Email, Telefono)
739 VALUES
740     ('Juan', 'Pérez', 'juan.perez@email.com', '987654321'),
741     ('María', 'González', 'maria.gonzalez@email.com', '987654322'),
742     ('Carlos', 'Rodríguez', 'carlos.rodriguez@email.com', '987654323');
743 END
744 GO

```

33 %

Messages

(3 rows affected)

Completion time: 2025-11-06T10:11:08.9718249-05:00

## CREACIÓN DE LA TABLA AUDITORIACLIENTES

```
746  -- a) Crear tabla AuditoriaClientes para registrar cambios
747  IF OBJECT_ID('AuditoriaClientes', 'U') IS NOT NULL
748  DROP TABLE AuditoriaClientes;
749  GO
750
751 CREATE TABLE AuditoriaClientes (
752     AuditoriaID INT PRIMARY KEY IDENTITY(1,1),
753     ClienteID INT NOT NULL,
754     TipoOperacion VARCHAR(10) NOT NULL, -- INSERT, UPDATE, DELETE
755     CampoModificado VARCHAR(50),
756     ValorAnterior NVARCHAR(MAX),
757     ValorNuevo NVARCHAR(MAX),
758     FechaModificacion DATETIME DEFAULT GETDATE(),
759     UsuarioModificacion NVARCHAR(100) DEFAULT SYSTEM_USER,
760     NombreHost VARCHAR(100) DEFAULT HOST_NAME(),
761     AplicacionOrigen NVARCHAR(128) DEFAULT APP_NAME()
762 );
763 GO
```

3 %

Messages

Commands completed successfully.

Completion time: 2025-11-06T13:12:09.5315525-05:00

```
765  -- Crear índice para consultas de auditoría
766  CREATE INDEX IX_AuditoriaClientes_ClienteID_Fecha
767  ON AuditoriaClientes(ClienteID, FechaModificacion DESC);
768  GO
769
770  CREATE INDEX IX_AuditoriaClientes_Usuario
771  ON AuditoriaClientes(UsuarioModificacion, FechaModificacion DESC);
772  GO
773
774  PRINT 'Tabla AuditoriaClientes creada exitosamente';
775  GO
776
```

100 %

Messages

Tabla AuditoriaClientes creada exitosamente

Completion time: 2025-11-06T13:13:32.3414447-05:00

## CREACIÓN DEL TRIGGER

```

//b
777  -- b) Crear trigger que registre en AuditoriaClientes cada eliminación
778  IF OBJECT_ID('TR_Clientes_Auditoria_Delete', 'TR') IS NOT NULL
779  DROP TRIGGER TR_Clientes_Auditoria_Delete;
780  GO
781
782  CREATE TRIGGER TR_Clientes_Auditoria_Delete
783  ON Clientes
784  AFTER DELETE
785  AS
786  BEGIN
787      SET NOCOUNT ON;
788
789      -- Insertar registro de auditoría por cada cliente eliminado
790      INSERT INTO AuditoriaClientes (
791          ClienteID,
792          TipoOperacion,
793          CampoModificado,
794          ValorAnterior,
795          ValorNuevo,
796          FechaModificacion,
797          UsuarioModificacion
798      )
799      SELECT
800          d.ClienteID,
801          'DELETE',
802          'REGISTRO_COMPLETO',
803          CONCAT('Nombre: ', d.Nombre, ', Apellido: ', d.Apellido, |
804                  ', Email: ', d.Email, ', Telefono: ', d.Telefono),
805          NULL,
806          GETDATE(),
807          SYSTEM_USER
808      FROM deleted d;
809
810      PRINT 'Auditoría de eliminación registrada';
811  END
812  GO
813

```

58 %

Messages

Commands completed successfully.

Completion time: 2025-11-06T13:14:11.7809561-05:00

## VERIFICACIÓN

```

915  -- Verificar trigger creado
916  SELECT
917      name AS NombreTrigger,
918      OBJECT_NAME(parent_id) AS TablaAsociada,
919      type_desc AS TipoTrigger,
920      is_disabled AS Desabilitado
921  FROM sys.triggers
922  WHERE parent_id = OBJECT_ID('Clientes');
923  GO

```

91 %

Results

	NombreTrigger	TablaAsociada	TipoTrigger	Deshabilitado
1	TR_Clientes_Auditoria_Delete	Clientes	SQL_TRIGGER	0

## JUSTIFICACIÓN TÉCNICA:

- Triggers AFTER: Ejecutan después de operaciones DML (INSERT, UPDATE, DELETE)

- Tablas inserted/deleted: Disponibles dentro de triggers
  - \* inserted: Contiene nuevos valores (INSERT, UPDATE)
  - \* deleted: Contiene valores anteriores (UPDATE, DELETE)
- Auditoría campo por campo: Permite rastrear exactamente qué cambió

### **INFORMACIÓN CAPTURADA:**

1. Qué cambió: TipoOperacion, CampoModificado
2. Cuándo cambió: FechaModificacion
3. Quién lo cambió: UsuarioModificacion (SYSTEM\_USER)
4. Desde dónde: NombreHost, AplicacionOrigen
5. Valores: ValorAnterior, ValorNuevo

### **BUENAS PRÁCTICAS:**

- ✓ Usar SET NOCOUNT ON en triggers para rendimiento
- ✓ Mantener lógica de triggers simple y rápida
- ✓ Considerar impacto en rendimiento de auditoría completa
- ✓ Implementar retención de datos de auditoría (ej: 7 años)
- ✓ Proteger tabla de auditoría con permisos restrictivos
- ✓ Considerar Change Data Capture (CDC) para tablas muy activas

#### **Proyecto 10: Simulación de restauración tras un incidente en QhatuPeru**

"Por un error humano se eliminó información de la tabla Clientes. El equipo debe restaurar la base usando el respaldo más reciente y validar la recuperación exitosa de los datos."

#### **Ejercicio práctico:**

- a) Simula la eliminación de registros de Clientes y la posterior restauración desde backup.
- b) Verifica la existencia y correcta restauración de los datos con una consulta.

## SCRIPT EN T-SQL:

```
994 USE master;
995 GO
996
997 -- ===== ESCENARIO: Simulación de incidente =====
998
999 PRINT '=====';
1000 PRINT 'SIMULACIÓN DE INCIDENTE Y RECUPERACIÓN';
1001 PRINT '=====';
1002 GO
1003
1004 -- Paso 1: Crear backup completo ANTES del incidente
1005 PRINT '';
1006 PRINT 'Paso 1: Creando backup completo antes del incidente...';
1007 GO
1008
1009 BACKUP DATABASE QhatuPeru
1010 TO DISK = 'D:\SQLBackups\QhatuPeru_PreIncidente.bak'
1011 WITH
1012   FORMAT,
1013   INIT,
1014   NAME = 'QhatuPeru-Backup Pre-Incidente',
1015   COMPRESSION,
1016   STATS = 10;
1017 GO
```

33 %

### Messages

```
=====
SIMULACIÓN DE INCIDENTE Y RECUPERACIÓN
=====

Paso 1: Creando backup completo antes del incidente...
18 percent processed.
36 percent processed.
54 percent processed.
72 percent processed.
81 percent processed.
91 percent processed.
100 percent processed.
Processed 704 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 24 pages for database 'QhatuPeru', file 'QhatuPeru_Secondary' on file 1.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP DATABASE successfully processed 730 pages in 0.056 seconds (101.771 MB/sec).

Completion time: 2025-11-06T13:20:58.0896398-05:00
```

```
1010
1019 -- Paso 2: Verificar datos actuales
1020 PRINT '';
1021 PRINT 'Paso 2: Datos actuales en tabla Clientes:';
1022 SELECT * FROM QhatuPeru.dbo.Clientes;
1023 GO
1024
```

33 %

### Results

	ClienteID	Nombre	Apellido	Email	Teléfono	FechaRegistro	Activo
1	1	Juan	Pérez	juan.perez@email.com	987654321	2025-11-06 13:11:08.970	1
2	2	Maria	González	maria.gonzalez@email.com	987654322	2025-11-06 13:11:08.970	1
3	3	Carlos	Rodríguez	carlos.rodriguez@email.com	987654323	2025-11-06 13:11:08.970	1

```

1024
1025    -- Paso 3: SIMULAR INCIDENTE - Eliminación accidental
1026    PRINT '';
1027    PRINT 'Paso 3: SIMULANDO INCIDENTE - Eliminación accidental de datos...';
1028    USE QhatuPeru;
1029    GO
1030
1031    -- Guardar conteo antes de eliminación
1032    DECLARE @ConteoAntes INT;
1033    SELECT @ConteoAntes = COUNT(*) FROM Clientes;
1034    PRINT 'Registros antes del incidente: ' + CAST(@ConteoAntes AS VARCHAR);
1035    GO
1036

```

83 %

#### Messages

Registros antes del incidente: 3

Completion time: 2025-11-06T13:23:19.3314563-05:00

```

1037    -- a) Simular eliminación accidental
1038    BEGIN TRANSACTION;
1039        DELETE FROM Clientes WHERE ClienteID IN (1, 2);
1040        PRINT 'ERROR SIMULADO: Eliminación accidental de registros críticos';
1041    COMMIT TRANSACTION;
1042    GO
1043
1044    -- Verificar datos después del incidente
1045    DECLARE @ConteoDespues INT;
1046    SELECT @ConteoDespues = COUNT(*) FROM Clientes;
1047    PRINT 'Registros después del incidente: ' + CAST(@ConteoDespues AS VARCHAR);
1048    GO
1049

```

83 %

#### Messages

ERROR SIMULADO: Eliminación accidental de registros críticos

Completion time: 2025-11-06T13:24:12.0590837-05:00

```

1050    SELECT * FROM Clientes;
1051    GO
1052
1053    PRINT '';
1054    PRINT '|ALERTA! Se detectó pérdida de datos. Iniciando proceso de recuperación...';
1055    GO
1056

```

83 %

#### Results

	ClienteID	Nombre	Apellido	Email	Teléfono	FechaRegistro	Activo
1	3	Carlos	Rodriguez	carlos.rodriguez@email.com	987654323	2025-11-06 13:11:08.970	1

```
1053 PRINT '';
1054 PRINT '|ALERTA! Se detectó pérdida de datos. Iniciando proceso de recuperación...';
1055 GO
1056
```

83 % ▾

Messages

|ALERTA! Se detectó pérdida de datos. Iniciando proceso de recuperación...

Completion time: 2025-11-06T13:25:17.5736795-05:00

```
1057 -- ===== PROCESO DE RESTAURACIÓN =====
1058
1059 -- Paso 4: Preparar restauración
1060 PRINT '';
1061 PRINT 'Paso 4: Preparando restauración desde backup...';
1062 GO
1063
1064 USE master;
1065 GO
1066
1067 -- Forzar cierre de conexiones activas
1068 ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
1069 GO
1070
```

33 % ▾

Messages

Paso 4: Preparando restauración desde backup...

Completion time: 2025-11-06T13:25:38.6587647-05:00

```
1071 -- Paso 5: Restaurar desde backup
1072 PRINT '';
1073 PRINT 'Paso 5: Ejecutando restauración desde backup pre-incidente...';
1074 GO
1075
1076 RESTORE DATABASE QhatuPeru
1077 FROM DISK = 'D:\SQLBackups\QhatuPeru_PreIncidente.bak'
1078 WITH
1079     REPLACE,
1080     RECOVERY,
1081     STATS = 10;
1082 GO
1083
```

3 % ▾

Messages

```
Paso 5: Ejecutando restauración desde backup pre-incidente...
18 percent processed.
36 percent processed.
54 percent processed.
72 percent processed.
90 percent processed.
100 percent processed.
Processed 704 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 24 pages for database 'QhatuPeru', file 'QhatuPeru_Secondary' on file 1.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
RESTORE DATABASE successfully processed 730 pages in 0.111 seconds (51.344 MB/sec).
```

Completion time: 2025-11-06T13:26:04.2233369-05:00

```

1084    -- Volver a modo multiusuario
1085    ALTER DATABASE QhatuPeru SET MULTI_USER;
1086    GO
1087
1088    PRINT 'Restauración completada exitosamente';
1089    GO
1090

```

83 %

### Messages

Restauración completada exitosamente

Completion time: 2025-11-06T13:26:28.9603347-05:00

```

1091    -- b) Verificar restauración correcta de los datos
1092    PRINT '';
1093    PRINT 'Paso 6: Verificando integridad de datos restaurados...';
1094    GO
1095
1096    USE QhatuPeru;
1097    GO
1098
1099    -- Contar registros después de restauración
1100    DECLARE @ConteoRestaurado INT;
1101    SELECT @ConteoRestaurado = COUNT(*) FROM Clientes;
1102    PRINT 'Registros después de restauración: ' + CAST(@ConteoRestaurado AS VARCHAR);
1103    GO
1104

```

3 %

### Messages

Paso 6: Verificando integridad de datos restaurados...  
Registros después de restauración: 3

Completion time: 2025-11-06T13:27:15.1981320-05:00

```

1105    -- Mostrar datos restaurados
1106    SELECT
1107        ClienteID,
1108        Nombre,
1109        Apellido,
1110        Email,
1111        Telefono,
1112        FechaRegistro
1113    FROM Clientes
1114    ORDER BY ClienteID;
1115    GO
1116

```

83 %

### Results

	ClienteID	Nombre	Apellido	Email	Telefono	FechaRegistro
1	1	Juan	Pérez	juan.perez@email.com	987654321	2025-11-06 13:11:08.970
2	2	Maria	González	maria.gonzalez@email.com	987654322	2025-11-06 13:11:08.970
3	3	Carlos	Rodriguez	carlos.rodriguez@email.com	987654323	2025-11-06 13:11:08.970

```

1123  -- Consultar historial de respaldos
1124  PRINT '';
1125  PRINT 'Paso 8: Historial de respaldos disponibles:';
1126  GO
1127
1128  SELECT
1129    database_name AS BaseDatos,
1130    backup_start_date AS FechaInicio,
1131    backup_finish_date AS FechaFin,
1132    type AS TipoBackup,
1133    CASE type
1134      WHEN 'D' THEN 'Completo'
1135      WHEN 'I' THEN 'Diferencial'
1136      WHEN 'L' THEN 'Log'
1137    END AS DescripcionTipo,
1138    backup_size / 1024 / 1024 AS TamañoMB,
1139    compressed_backup_size / 1024 / 1024 AS TamañoComprimidoMB,
1140    physical_device_name AS RutaArchivo
1141  FROM msdb.dbo.backupset bs
1142  INNER JOIN msdb.dbo.backupmediafamily bmf ON bs.media_set_id = bmf.media_set_id
1143  WHERE database_name = 'QhatuPeru'
1144  ORDER BY backup_start_date DESC;
1145  GO
1146

```

83 %

	BaseDatos	FechaInicio	FechaFin	TipoBackup	DescripcionTipo	TamañoMB	TamañoComprimidoMB	RutaArchivo
1	QhatuPeru	2025-11-06 13:20:57.000	2025-11-06 13:20:58.000	D	Completo	7.08593750000	1.99465751562	D:\SQLBackups\QhatuPeru_PreIncidente.bak
2	QhatuPeru	2025-11-06 12:25:07.000	2025-11-06 12:25:07.000	I	Diferencial	2.08593750000	0.14874839746	D:\SQLBackups\QhatuPeru_Diff.bak
3	QhatuPeru	2025-11-06 12:25:07.000	2025-11-06 12:25:07.000	L	Log	0.14062500000	0.01506519238	D:\SQLBackups\QhatuPeru_Log.trn
4	QhatuPeru	2025-11-06 12:25:03.000	2025-11-06 12:25:03.000	D	Completo	6.08593750000	1.60775089257	D:\SQLBackups\QhatuPeru_Full.bak

## JUSTIFICACIÓN TÉCNICA:

- RESTORE DATABASE: Recupera BD desde archivo de backup
- WITH REPLACE: Sobrescribe BD existente
- WITH RECOVERY: Deja BD lista para uso (estado ONLINE)
- SINGLE\_USER: Necesario para evitar conflictos durante restauración
- DBCC CHECKDB: Verifica integridad física y lógica de la BD

## TIPOS DE RESTAURACIÓN:

### 1. Restauración completa (FULL):

- Restaura toda la base de datos
- Punto de recuperación: momento del backup

### 2. Restauración point-in-time:

- Requiere backup FULL + backups de LOG
- Permite restaurar a momento específico
- Sintaxis: RESTORE LOG ... WITH STOPAT = '2024-01-15 14:30:00'

### 3. Restauración por páginas:

- Restaura solo páginas dañadas
- BD permanece online durante proceso
- Requiere modelo de recuperación FULL

### **SECUENCIA COMPLETA DE RECUPERACIÓN:**

1. Backup FULL más reciente
2. Backup DIFERENCIAL más reciente (si existe)
3. Todos los backups de LOG desde el diferencial
4. RESTORE WITH RECOVERY en el último paso

### **BUENAS PRÁCTICAS:**

- ✓ Realizar backups antes de cambios importantes
- ✓ Probar restauraciones regularmente (no solo backups)
- ✓ Documentar RTO (Recovery Time Objective)
- ✓ Mantener backups en múltiples ubicaciones
- ✓ Automatizar alertas de backups fallidos
- ✓ Calcular RPO (Recovery Point Objective) aceptable
- ✓ Entrenar al equipo en procedimientos de recuperación