

RESPALDO Y RECUPERACION DE DATOS

Proyecto 1: Estrategia de backup completo diario y verificación automática

1. Enunciado del ejercicio

Implementar un script que haga un **backup FULL diario** de la base QhatuPeru, almacene el archivo con fecha, y verifique la integridad del backup (RESTORE VERIFYONLY). Programar (demostración por script) un job de SQL Agent que ejecute la operación.

El enunciado pide:

1. Implementar un script que realice un **backup FULL diario** de la base de datos QhatuPeru.
2. Almacenar el archivo de backup con la fecha.
3. Verificar la integridad del backup usando RESTORE VERIFYONLY.
4. Programar un **Job de SQL Agent** para ejecutar esta operación diariamente.

A continuación, te presento el código SQL resumido para el script de backup y verificación, seguido de los pasos para configurar el Job en SQL Agent.

1. Script SQL para Backup Completo y Verificación

Este script utiliza variables para la fecha y la ruta del archivo, lo que facilita su uso dentro de un Job de SQL Agent.

```
1  -- Cambiar al contexto de la base de datos master
2  USE [master];
3  GO
4
5  -- Declaración de variables
6  DECLARE @BackupPath NVARCHAR(1000) = N'C:\SQLBackups\'; -- **VERIFICA LA RUTA**
7  DECLARE @DatabaseName SYSNAME = N'QhatuPeru';
8  DECLARE @FileName NVARCHAR(256);
9  DECLARE @BackupFilePath NVARCHAR(1256);
10 DECLARE @FechaBackup NVARCHAR(8);
11 DECLARE @BackupSetName NVARCHAR(128); -- Variable para evitar el error de sintaxis '+'
12
13 -- 1. Capturar y formatear la fecha (YYYYMMDD)
14 SET @FechaBackup = CONVERT(NVARCHAR(8), GETDATE(), 112);
15
16 -- 2. Construir el nombre completo del archivo
17 SET @FileName = @DatabaseName + N'_FULL_' + @FechaBackup + N'.bak';
18 SET @BackupFilePath = @BackupPath + @FileName;
19
20 -- 3. Pre-concatenar el nombre del conjunto de backup (CORRIGE ERROR 102)
21 SET @BackupSetName = @DatabaseName + N' Full Backup ' + @FechaBackup;
22
23 PRINT N'Iniciando Backup para: ' + @DatabaseName;
```

2. Configuración del Job de SQL Agent

Para programar esta operación diariamente, debes crear un Job en SQL Server Agent:

Paso 1: Crear el Job

1. Abre **SQL Server Management Studio (SSMS)**.
2. Expande **SQL Server Agent**.
3. Haz clic derecho en **Jobs** y selecciona **New Job...**
4. En la pestaña **General**, dale un nombre significativo, por ejemplo:
Daily_FULL_Backup_QhatuPeru.

Paso 2: Crear el Paso de Ejecución (Step)

1. Ve a la pestaña **Steps** y haz clic en **New...**
2. En **Step name**, ponle un nombre (ej: Backup_Verify_Script).
3. En **Type**, selecciona **Transact-SQL script (TSQL)**.
4. En **Database**, selecciona la base de datos donde se ejecutará (generalmente master o msdb, ya que el script maneja la base de datos por nombre).
5. Copia y pega el código SQL de la sección 1 en el campo **Command**.

Paso 3: Configurar la Programación (Schedules)

1. Ve a la pestaña **Schedules** y haz clic en **New...**
2. En **Name**, ponle un nombre (ej: Daily_Midnight).
3. En **Schedule Type**, selecciona **Recurring**.
4. En **Frequency**:
 - **Occurs:** Daily
 - **Recurs every:** 1 day
5. En **Daily Frequency**:
 - **Occurs once at:** Elige la hora deseada para la ejecución diaria (ej: 1:00:00 AM).

```
22
23 PRINT N'Iniciando Backup para: ' + @DatabaseName;
24 PRINT N'Ruta del archivo de backup: ' + @BackupFileFullPath;
25
26 -- 4. Realizar el Backup FULL (CORRIGE ERROR 1944 - COMPRESSION NO SOPORTADA)
27 BACKUP DATABASE @DatabaseName
28 TO DISK = @BackupFileFullPath
29 WITH NOINIT,
30     NAME = @BackupSetName, -- Se usa la variable pre-concatenada
31     STATS = 10;
32
33 PRINT N'Backup realizado. Iniciando verificación...';
34
35 -- 5. Verificar la integridad del Backup (RESTORE VERIFYONLY)
36 RESTORE VERIFYONLY
37 FROM DISK = @BackupFileFullPath;
38
39 -- El error de archivo no encontrado (3201) se corrige automáticamente al tener un backup exitoso.
40
41 PRINT N'Backup de QhatuPeru realizado y verificado con éxito.';
```

83 % 1 0 ↑ ↓ Línea: 41, Carácter: 64 SPC CRLF Windows

Mensajes

```
Iniciando Backup para: QhatuPeru
Ruta del archivo de backup: C:\SQLBackups\QhatuPeru_FULL_20251210.bak
10 percent processed.
21 percent processed.
31 percent processed.
40 percent processed.
50 percent processed.
61 percent processed.
70 percent processed.
80 percent processed.
91 percent processed.
100 percent processed.
Processed 464 pages for database 'QhatuPeru', file 'QhatuPeru' on file 1.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_log' on file 1.
BACKUP DATABASE successfully processed 466 pages in 0.026 seconds (129.878 MB/sec).
Backup realizado. Iniciando verificación...
The backup set on file 1 is valid.
Backup de QhatuPeru realizado y verificado con éxito.

Hora de finalización: 2025-12-10T21:18:14.4199235-05:00
```

Proyecto 2 — Estrategia completa: backup full semanal + verificación automática

2. Enunciado del ejercicio

Implementar un plan que realice un **backup completo** de la base QhatuPERU cada domingo a las 02:00, compruebe la validez del backup y registre el resultado en una tabla de auditoría.

Implementación de la Tabla de Auditoría

Primero, crea una tabla para registrar los resultados del backup. Ejecuta esto una sola vez, en la base de datos QhatuPeru o en msdb

2. Script SQL para Backup Semanal y Auditoría

Este script está corregido (sin COMPRESSION ni el error +) e incluye el registro del proceso en la tabla BackupAuditLog.

Configuración del Job Semanal en SQL Agent

Para programar la ejecución del script (Punto 2) cada domingo a las 02:00:

Paso 1: Crear el Job

1. Crea un nuevo Job, nómbralo, por ejemplo, Weekly_FULL_Audit_QhatuPeru.

Paso 2: Crear el Paso de Ejecución (Step)

1. Crea un nuevo Step.
2. **Tipo: Transact-SQL script (TSQL)** (Esto previene el error 2812).
3. **Database:** master o msdb.
4. **Command:** Pega el **Código SQL** completo del Punto 2.

Paso 3: Configurar la Programación (Schedule)

1. Crea un nuevo Schedule.
2. **Frequency:**
 - **Occurs:** Weekly
 - **Recurs every:** 1 week(s)
 - **On:** Selecciona **Domingo**
3. **Daily Frequency:**
 - **Occurs once at:** 02:00:00 AM

```

1  USE [QhatuPeru]; -- Puedes usar msdb si prefieres centralizar logs
2  GO
3
4  CREATE TABLE dbo.BackupAuditLog (
5      LogID INT IDENTITY(1,1) PRIMARY KEY,
6      DatabaseName SYSNAME NOT NULL,
7      BackupStartTime DATETIME NOT NULL,
8      BackupEndTime DATETIME NULL,
9      BackupStatus VARCHAR(50) NOT NULL, -- Ej: 'EXITOSO', 'FALLIDO'
10     VerifyStatus VARCHAR(50) NULL,      -- Ej: 'OK', 'FALLIDO'
11     FileName NVARCHAR(500) NULL,
12     ErrorMessage NVARCHAR(MAX) NULL    -- Para registrar cualquier error
13 );
14 GO
15 USE [master];
16 GO
17
-- Variables de Configuración
DECLARE @DatabaseName SYSNAME = N'QhatuPeru';
DECLARE @BackupPath NVARCHAR(1000) = N'C:\SQLBackups\'; -- **VERIFICA ESTA RUTA**
DECLARE @BackupType VARCHAR(10) = 'FULL';
DECLARE @LogID INT;
DECLARE @BackupStartDate DATETIME = GETDATE();

-- Variables de Script
DECLARE @FileName NVARCHAR(256);
DECLARE @BackupFileFullPath NVARCHAR(1256);
DECLARE @FechaBackup NVARCHAR(8) = CONVERT(NVARCHAR(8), GETDATE(), 112);
DECLARE @BackupSetName NVARCHAR(128);

-- 1. Iniciar registro de Auditoria y capturar el ID de Log
INSERT INTO [QhatuPeru].dbo.BackupAuditLog (DatabaseName, BackupStartTime, BackupStatus)
VALUES (@DatabaseName, @BackupStartDate, 'EN PROGRESO');

SET @LogID = SCOPE_IDENTITY();

BEGIN TRY
    -- 2. Construir rutas y nombres
    SET @FileName = @DatabaseName + N'_' + @BackupType + N'_' + @FechaBackup + N'.bak';
    SET @BackupFileFullPath = @BackupPath + @FileName;
    SET @BackupSetName = @DatabaseName + N' Weekly Full Backup ' + @FechaBackup;

    -- 3. Ejecutar Backup (Sin COMPRESSION para evitar error 1944/3013 en Express)
    BACKUP DATABASE @DatabaseName
    TO DISK = @BackupFileFullPath
    WITH NOINIT,
        NAME = @BackupSetName,
        STATS = 10;

    -- 4. Verificar Integridad
    RESTORE VERIFYONLY
    FROM DISK = @BackupFileFullPath;

    -- 5. Actualizar registro de Auditoria (EXITO)
    UPDATE [QhatuPeru].dbo.BackupAuditLog
    SET BackupEndTime = GETDATE(),
        BackupStatus = 'EXITOSO',
        VerifyStatus = 'OK',
        FileName = @BackupFileFullPath
    WHERE LogID = @LogID;

END TRY
BEGIN CATCH
    -- 6. Manejo de errores y registro de FALLO
    DECLARE @ErrorMessage NVARCHAR(4096) = ERROR_MESSAGE();

    UPDATE [QhatuPeru].dbo.BackupAuditLog
    SET BackupEndTime = GETDATE(),
        BackupStatus = 'FALLIDO',
        VerifyStatus = 'NO APLICÓ',
        FileName = @BackupFileFullPath,
        ErrorMessage = @ErrorMessage
    WHERE LogID = @LogID;

    -- Opcional: Levantar el error para que el Job de SQL Agent falle
    THROW;
END CATCH
GO

```

```

(1 fila afectada)
10 percent processed.
20 percent processed.
30 percent processed.
40 percent processed.
50 percent processed.
60 percent processed.
70 percent processed.
80 percent processed.
90 percent processed.
100 percent processed.
Processed 528 pages for database 'QhatuPeru', file 'QhatuPeru' on file 2.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_log' on file 2.
BACKUP DATABASE successfully processed 530 pages in 0.030 seconds (137.890 MB/sec).
The backup set on file 1 is valid.

```

EXITOSO/OK: Si el Job se ejecuta correctamente, verás 'EXITOSO' en Estado Backup y 'OK' en Estado Verificación.

FALLIDO: Si el Job falla, verás 'FALLIDO' en Estado Backup y el campo Mensaje de Error contendrá el error de SQL que causó la terminación anormal del proceso.

	LogID	Base de Datos	Inicio Backup	Fin Backup	Duración (min)	Estado Backup	Estado Verificación	Archivo Generado	Mensaje de Error
1	1	QhatuPeru	2025-12-10 21:22:58.133	2025-12-10 21:22:58.243	0	EXITOSO	OK	C:\SQLBackups\QhatuPeru_FULL_20251210.bak	NULL

Proyecto 3: Estrategia combinada: FULL semanal + DIFF diario + LOG cada 30 minutos

1. Enunciado del ejercicio

Diseñar scripts que implementen: backup FULL semanal, backup diferencial diario (entre FULLs) y backup de transacciones (LOG) cada 30 minutos. Demostrar cómo restaurar una cadena (FULL + última DIFF + logs).

1. Preparación Inicial de la Base de Datos

Antes de cualquier backup diferencial o de log, debes asegurarte de que la base de datos QhatuPeru cumpla con dos requisitos:

1. Estar en el **Modo de Recuperación Completo (FULL Recovery Model)**.
2. Tener al menos un **Backup FULL inicial**.

2. Scripts para los Jobs de SQL Agent

Necesitarás **tres Jobs distintos** en SQL Agent (uno para cada tipo de backup). Recuerda configurar el **Tipo** de cada Step como **Transact-SQL script (TSQL)** para evitar el error Mens. 2812.

A. Job 1: FULL Semanal (Domingo 02:00)

Este establece la base semanal para el diferencial.

B. Job 2: Diferencial Diario (Lunes a Sábado 02:00)

Este solo copia los cambios desde el último FULL.

C. Job 3: Log de Transacciones (Cada 30 minutos)

Este copia los cambios desde el último LOG o FULL/DIFF. **Crucial:** Debe ser ejecutado con frecuencia para mantener el tamaño del archivo de log.

3. Demostración de Restauración de Cadena (Recovery Chain)

La secuencia de restauración sigue un orden estricto: **FULL \$ \rightarrow \$ DIFF \$ \rightarrow \$ LOGs**. La clave es usar NORECOVERY en los backups intermedios y RECOVERY solo en el último LOG.

Escenario de Ejemplo

- **FULL** Semanal: QhatuPeru_FULL_20251207.bak
- **DIFF** Diario (Último): QhatuPeru_DIFF_20251210.dif
- **LOGs** (Necesarios): QhatuPeru_LOG_20251210_0900.trn, QhatuPeru_LOG_20251210_0930.trn, etc.

-- 1. Restaurar el último FULL semanal (Base de la cadena)

```
RESTORE DATABASE [QhatuPeru_Recovered]
```

```
FROM DISK = N'C:\SQLBackups\QhatuPeru_FULL_20251207.bak'
```

```
WITH REPLACE, NORECOVERY; -- NORECOVERY es VITAL aquí
```

-- 2. Restaurar el último DIFF (Solo los cambios desde el FULL)

```
RESTORE DATABASE [QhatuPeru_Recovered]
```

```
FROM DISK = N'C:\SQLBackups\QhatuPeru_DIFF_20251210.dif'
```

```
WITH NORECOVERY; -- NORECOVERY es VITAL aquí
```

-- 3. Aplicar todos los LOGs en orden cronológico

-- (Se aplica el primer log después del DIFF)

```
RESTORE LOG [QhatuPeru_Recovered]
```

```
FROM DISK = N'C:\SQLBackups\QhatuPeru_LOG_20251210_0900.trn'
```

```
WITH NORECOVERY;
```

-- (Se aplica el siguiente log)

```
RESTORE LOG [QhatuPeru_Recovered]
```

```
FROM DISK = N'C:\SQLBackups\QhatuPeru_LOG_20251210_0930.trn'
```

```
WITH NORECOVERY;
```

-- ... (Continuar con todos los logs intermedios)

-- 4. Aplicar el ÚLTIMO LOG y poner la base de datos en estado operativo (RECOVERY)

RESTORE LOG [QhatuPeru_Recovered]

FROM DISK = N'C:\SQLBackups\QhatuPeru_LOG_20251210_1000.trn'

WITH RECOVERY; -- RECOVERY es VITAL en el último paso

```
59
60 USE [master];
61 GO
62 DECLARE @DB SYSNAME = N'QhatuPeru', @Path NVARCHAR(1000) = N'C:\SQLBackups\';
63 DECLARE @TS NVARCHAR(15) = REPLACE(REPLACE(CONVERT(NVARCHAR(19), GETDATE(), 120), '-', ''), ':', ''); -- Timestamp sin caracteres especiales
64 DECLARE @File NVARCHAR(1256) = @Path + @DB + N'_LOG_' + @TS + N'.trn';
65 DECLARE @BackupSetName NVARCHAR(128) = @DB + N' T-Log ' + @TS; -- Concatenación corregida
66
67 BACKUP LOG @DB
68 TO DISK = @File
69 WITH NOINIT, NAME = @BackupSetName, STATS = 10;
70 GO
71
```

91 % 1 0 LÍNEA: 71, CARÁCTER: 1 SPC CRLF Windows

Mensajes

26 percent processed.
52 percent processed.
78 percent processed.
100 percent processed.
Processed 31 pages for database 'QhatuPeru', file 'QhatuPeru_log' on file 1.
BACKUP LOG successfully processed 31 pages in 0.005 seconds (47.656 MB/sec).
Hora de finalización: 2025-12-10T21:56:32.2155933-05:00

Proyecto 4 — Estrategia diferencial: snapshot entre backups completos

2. Enunciado del ejercicio

Automatizar backups **diferenciales diarios** (Lun–Sáb) y conservarlos por 14 días. Implementar script de limpieza basado en msdb.dbo.backupset para eliminar archivos antiguos.

1. Implementación del Backup Diferencial Diario

Este script T-SQL debe ejecutarse todos los días, excepto el domingo (cuando se ejecuta el FULL Semanal).

Job Name: P4_DIFF_Diario_QhatuPeru

2. Implementación del Script de Limpieza (Retención de 14 Días)

El enunciado pide conservar los archivos por **14 días** e implementar un script de limpieza basado en msdb.dbo.backupset. La forma más eficiente de limpiar los archivos físicos en SQL Server es mediante xp_delete_files, y para la historia de msdb es sp_delete_backuphistory.

B. T-SQL para Limpieza del Historial (msdb.dbo.backupset)

Esta es la forma oficial de limpiar las tablas del sistema (msdb.dbo.backupset, etc.) para que no crezcan demasiado.


```

1  USE [master];
2  GO
3  -- Variables (usando QhatuPeru, como en Proyectos 1-3)
4  DECLARE @DB SYSNAME = N'QhatuPeru',
5           @Path NVARCHAR(1000) = N'C:\SQLBackups\',
6           @Fecha NVARCHAR(8) = CONVERT(NVARCHAR(8), GETDATE(), 112);
7
8  -- Crear nombres de archivo y de conjunto de backup (resuelve Mens. 102)
9  DECLARE @File NVARCHAR(1256) = @Path + @DB + N'_DIFF_' + @Fecha + N'.dif';
10 DECLARE @BackupSetName NVARCHAR(128) = @DB + N' Daily DIFF ' + @Fecha;
11
12 -- Ejecutar el Backup Diferencial
13 BACKUP DATABASE @DB
14 TO DISK = @File
15 WITH NOINIT,
16     DIFFERENTIAL, -- CLAVE para este proyecto
17     NAME = @BackupSetName,
18     STATS = 10;
19 GO

```

91 % No se encontraron problemas.

Línea: 19, Carácter: 3 SPC

Mensajes

```

14 percent processed.
22 percent processed.
36 percent processed.
44 percent processed.
51 percent processed.
66 percent processed.
73 percent processed.
80 percent processed.
95 percent processed.
100 percent processed.
Processed 120 pages for database 'QhatuPeru', file 'QhatuPeru' on file 2.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_log' on file 2.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 122 pages in 0.018 seconds (52.734 MB/sec).

Hora de finalización: 2025-12-10T21:43:58.2758485-05:00

```

SQLQuery9.s...elvis (59))* SQLQuery8.sql...P\elvis (76))* SQLQuery7.sql...P\elvis (84))* SQLQuery6.sql...P\elvis (56))* SQLQuery5.sql...P\elvis (55))*

```

1  -- Este es el código que se ejecutará correctamente una vez que los permisos estén listos:
2  USE [master];
3  GO
4  DECLARE @DB SYSNAME = N'QhatuPeru', @Path NVARCHAR(1000) = N'C:\SQLBackups\', @Fecha NVARCHAR(8) = CONVERT(NVARCHAR(8), GETDATE(), 112);
5  DECLARE @File NVARCHAR(1256) = @Path + @DB + N'_FULL_' + @Fecha + N'.bak';
6  DECLARE @BackupSetName NVARCHAR(128) = @DB + N' Weekly FULL ' + @Fecha;
7
8  BACKUP DATABASE @DB
9  TO DISK = @File
10 WITH NOINIT, NAME = @BackupSetName, STATS = 10;
11 RESTORE VERIFYONLY FROM DISK = @File;
12 GO

```

100 % No se encontraron problemas.

Línea: 12, Carácter: 3 TABULACIONES CRLF Windows 1252

Mensajes

```

11 percent processed.
21 percent processed.
31 percent processed.
41 percent processed.
51 percent processed.
61 percent processed.
71 percent processed.
81 percent processed.
91 percent processed.
100 percent processed.
Processed 576 pages for database 'QhatuPeru', file 'QhatuPeru' on file 1.
Processed 2 pages for database 'QhatuPeru', file 'QhatuPeru_log' on file 1.
BACKUP DATABASE successfully processed 578 pages in 0.030 seconds (150.390 MB/sec).
The backup set on file 1 is valid.

Hora de finalización: 2025-12-10T22:03:00.8591166-05:00

```



```
SQLQuery9.s...lvis (59))* X SQLQuery8.sql...lvis (76))* SQLQuery7.sql...lvis (84))* SQLQuery6.sql...lvis (56))* SQLQuery5.sql...lvis (55))*
3
4 -- 1. Variables de configuración y cálculo de fecha
5 DECLARE @BackupPath NVARCHAR(1000) = N'C:\SQLBackups'; -- RUTA SIN BARRA FINAL
6 DECLARE @RetentionDays INT = 14;
7 DECLARE @DeleteDate NVARCHAR(10);
8 DECLARE @Command NVARCHAR(2000);
9
10 -- Calcular la fecha límite en formato YYYY-MM-DD (Formato 23, el más seguro)
11 SET @DeleteDate = CONVERT(NVARCHAR(10), DATEADD(dd, -@RetentionDays, GETDATE()), 23);
12
13 PRINT N'Fecha Límite para eliminación de archivos: ' + @DeleteDate;
14
15 -- 2. Limpieza del Historial de MSDB (Requisito del Proyecto 4)
16 -- Esto elimina las entradas en msdb.dbo.backupset más antiguas que 14 días.
17 EXEC dbo.sp_delete_backuphistory @oldest_date = @DeleteDate;
18 PRINT N'Historial de MSDB limpiado.';
19
20 -- 3. Limpieza de Archivos Físicos (FULL/DIFF - .bak)
```

100 % No se encontraron problemas. Línea: 32, Carácter: 3 (86 caracteres, 2 líneas) TABULACIONES CRLF

Mensajes

Limpieza física completada. Archivos más antiguos de 14 días eliminados.

Hora de finalización: 2025-12-10T22:04:56.8069109-05:00

Proyecto 5: Restauración en distintos escenarios: recuperación de corrupción parcial y recuperación de emergencia

1. Enunciado del ejercicio

Practicar restauraciones en 3 escenarios: a) restauración completa (recuperación estándar), b) restauración punto-en-el-tiempo por fallo humano (ejemplo: eliminación masiva accidental), c) restauración desde backup corrupto mediante verificación y usar backup alternativo.

A: Restauración Completa (Recuperación Estándar)

Objetivo: Restaurar la base de datos al momento más reciente posible (recuperación de emergencia).

Se utiliza la cadena completa (FULL \rightarrow DIFF \rightarrow LOGs) con RECOVERY en el último paso.

Escenario B: Restauración Punto-en-el-Tiempo (Fallo Humano)

Objetivo: Restaurar la base de datos a un momento **antes** de que ocurriera un fallo humano (ej., una eliminación masiva accidental).

Se utiliza la cláusula STOPAT = 'YYYY-MM-DD HH:MM:SS' para detener la restauración justo antes del desastre.

Escenario C: Restauración por Backup Corrupto (Verificación)

Objetivo: Simular que el backup primario está corrupto, lo que obliga a usar el backup anterior.

La herramienta clave es RESTORE VERIFYONLY para detectar la corrupción *antes* de restaurar.

Paso 1: Simular la Detección de Corrupción

Si el backup primario (el más reciente) falla la verificación, el Job debe pasar al backup alternativo.

Paso 2: Restaurar desde Backup Alternativo (Anterior)

Si la verificación falla, se usa el backup FULL anterior (alternativo) como base para la restauración.

