

1. Estrategias de Backup: Completo, Diferencial, Transaccional

Script T-SQL

```
1  -- 1. Configurar el modelo de recuperación a FULL para habilitar logs
2  ALTER DATABASE [QhatuPERU] SET RECOVERY FULL;
3  GO
4
5  -- 2. Backup Completo (Full) - Línea Base
6  BACKUP DATABASE [QhatuPERU]
7  TO DISK = 'D:\SQLBackups\QhatuPERU_FULL.bak'
8  WITH INIT, COMPRESSION, STATS = 10;
9  GO
10
11 -- 3. Backup Diferencial (Differential) - Cambios desde el último Full
12 BACKUP DATABASE [QhatuPERU]
13 TO DISK = 'D:\SQLBackups\QhatuPERU_DIFF.bak'
14 WITH DIFFERENTIAL, COMPRESSION, INIT, STATS = 10;
15 GO
16
17 -- 4. Backup de Log de Transacciones (Transaccional)
18 BACKUP LOG [QhatuPERU]
19 TO DISK = 'D:\SQLBackups\QhatuPERU_LOG_001.trn'
20 WITH NOINIT, COMPRESSION, STATS = 10;
21 GO
22
```

Justificación Técnica

La configuración del modelo **RECOVERY FULL** es fundamental, ya que garantiza que **todas las transacciones** se registren en el log, permitiendo la **restauración a un punto en el tiempo (PIT)** y minimizando la pérdida de datos (RPO bajo). El backup **Completo** es necesario para iniciar la cadena de respaldo. El **Diferencial** reduce el volumen de datos a copiar en los backups intermedios, y el **Transaccional** captura la actividad más reciente.

Buenas Prácticas

- **Ruta Externa:** Almacene los respaldos en una ubicación **distinta** a la de los archivos .mdf y .ldf de QhatuPERU.
- **Monitoreo:** Revise periódicamente el tamaño del log de transacciones (.ldf) en el modelo Full, ya que solo se trunca después de un backup de log exitoso.
- **Frecuencia:** Defina el Full semanalmente, el Diferencial diariamente y el Log cada 15-60 minutos, según la criticidad de los datos.

2. Restauración de bases de datos en distintos escenarios

Script T-SQL

```
24  -- Definir la ruta del archivo de backup Full
25  DECLARE @BackupPath NVARCHAR(500) = 'E:\SQLBackups\QhatuPERU_FULL.bak';
26
27  -- 1. Obtener los nombres lógicos de los archivos del backup
28  RESTORE FILELISTONLY
29  FROM DISK = @BackupPath;
30  -- Debería retornar algo como: LogicalName='QhatuPERU_Data' y LogicalName='QhatuPERU_Log'
31
32  -- 2. Restaurar el backup con la cláusula WITH MOVE para reubicar archivos
33  RESTORE DATABASE [QhatuPERU_MIGRACION] -- Nuevo nombre de BD
34  FROM DISK = @BackupPath
35  WITH MOVE 'QhatuPERU_Data' TO 'D:\SQL_Migration\QhatuPERU_MIGRACION.mdf', -- Nueva ruta para datos
36    MOVE 'QhatuPERU_Log' TO 'D:\SQL_Migration\QhatuPERU_MIGRACION_log.ldf', -- Nueva ruta para log
37    REPLACE, -- Sobrescribe si la BD ya existe
38    STATS = 10;
39  GO
40
```

Justificación Técnica

La cláusula **WITH MOVE** es esencial porque los archivos \$mdf\$ y \$ldf\$ en el backup contienen las rutas originales del servidor de origen. Sin esta cláusula, el proceso de restauración fallaría en el nuevo servidor si las rutas físicas no existen. El comando **RESTORE FILELISTONLY** es necesario para obtener los **nombres lógicos** de los archivos (QhatuPERU_Data, QhatuPERU_Log), que son los que se mencionan en la cláusula MOVE.

Buenas Prácticas

- **Verificación Previa:** Utilice siempre **RESTORE FILELISTONLY** para confirmar los nombres lógicos y evitar errores.
- **Seguridad:** Asegúrese de que la cuenta de servicio de SQL Server tenga permisos de **escritura** en las nuevas rutas físicas especificadas en la cláusula **MOVE**.
- **REPLACE:** Use **WITH REPLACE** solo cuando esté seguro de sobrescribir una base de datos existente con el mismo nombre en el destino.

3. Mantenimiento y verificación de copias de seguridad

Script T-SQL

```
41  -- 1. Backup Completo de QhatuPERU con Compresión y Checksum
42  BACKUP DATABASE [QhatuPERU]
43  TO DISK = 'D:\SQLBackups\QhatuPERU_FULL_CHK.bak'
44  WITH INIT,
45  [
46      **COMPRESSION**, -- Reduce tamaño del archivo y tiempo de I/O
47      **CHECKSUM**,    -- Agrega validación de integridad de página/escritura
48      STATS = 10;
49  GO
50
51  -- 2. Verificar la integridad y legibilidad del archivo de Backup
52  RESTORE **VERIFYONLY** FROM DISK = 'D:\SQLBackups\QhatuPERU_FULL_CHK.bak';
53  GO
54
```

Justificación Técnica

El uso de **WITH COMPRESSION** reduce el tamaño del archivo de backup, disminuyendo los tiempos de E/S y el espacio de almacenamiento. **WITH CHECKSUM** fuerza a SQL Server a calcular una suma de verificación para cada página y validar los datos al momento de la copia, previniendo que se almacenen datos corruptos. **RESTORE VERIFYONLY** comprueba que el motor pueda leer todo el contenido del archivo de respaldo sin iniciar una restauración real, confirmando su **recuperabilidad**.

Buenas Prácticas

- **Checks en Origen:** La opción **CHECKSUM** debe ser usada consistentemente para asegurar la calidad del backup en el momento de la creación.
- **Prueba de Rutina:** La verificación con **VERIFYONLY** debe ser parte de la rutina automatizada, ejecutándose inmediatamente después de la creación de cada archivo de respaldo.
- **Consistencia:** Asegúrese de ejecutar **DBCC CHECKDB** en la base de datos de origen **QhatuPERU** periódicamente para encontrar corrupción antes de que sea respaldada.

4. Uso de **BACKUP DATABASE** y **RESTORE DATABASE**

Script T-SQL

```

58      -- 1. Restaurar el Backup COMPLETO
59      RESTORE DATABASE [QhatuPERU]
60      FROM DISK = 'D:\SQLBackups\QhatuPERU_FULL.bak'
61      WITH **NORECOVERY**, -- Mantiene la BD en estado 'Restoring'
62          ~REPLACE,
63          STATS = 10;
64      GO
65
66
67      -- 2. Restaurar el Backup DIFERENCIAL
68      RESTORE DATABASE [QhatuPERU]
69      FROM DISK = 'D:\SQLBackups\QhatuPERU_DIFF.bak'
70      WITH **NORECOVERY**, -- Mantiene la BD en estado 'Restoring'
71          ~STATS = 10;
72      GO
73
74      -- 3. Restaurar el Backup de LOG FINAL y aplicar RECOVERY
75      RESTORE LOG [QhatuPERU]
76      FROM DISK = 'D:\SQLBackups\QhatuPERU_LOG_002.trn'
77      WITH **RECOVERY**, -- Finaliza la restauración y pone la BD ONLINE
78          ~STATS = 10;
79      GO
80

```

Justificación Técnica

El uso de **WITH NORECOVERY** es crucial en los pasos iniciales (Full y Differential) porque evita que SQL Server revierta las transacciones incompletas y, más importante, mantiene la base de datos en un estado que permite la aplicación de archivos de respaldo **adicionales** (los logs). El último comando de la secuencia debe usar **WITH RECOVERY** para completar el *roll forward* y *roll back* y dejar QhatuPERU accesible para los usuarios.

Buenas Prácticas

- **Identificación de Archivos:** Antes de una restauración de emergencia, use **RESTORE HEADERONLY** en los archivos para confirmar que está aplicando el Full correcto, el último Differential y la secuencia correcta de Logs.
- **Monitoreo del Progreso:** Use **STATS = 10** para obtener información del progreso de la restauración en intervalos del 10%.
- **Exclusividad:** Asegúrese de que no haya conexiones activas a la base de datos antes de iniciar la restauración.

5. Planes de mantenimiento y políticas de retención

Script T-SQL

```
85  -- Parámetros de la Política de Retención (7 días)
86  DECLARE @RetencionDias INT = 7;
87  DECLARE @FechaCorte DATETIME = DATEADD(dd, -@RetencionDias, GETDATE());
88  DECLARE @DirectorioBackup NVARCHAR(500) = N'D:\SQLBackups';
89
90  -- Uso del procedimiento para limpiar archivos antiguos
91  EXECUTE master.dbo.xp_delete_file
92      0,                                     -- Tipo de archivo (0=Backup)
93      @DirectorioBackup,                     -- Directorio de backups de QhatuPERU
94      N'bak',                                -- Extensión para Full y Diff
95      @FechaCorte,                           -- Fecha de corte (archivos más antiguos se borran)
96      1;                                     -- Incluir subcarpetas (1=Sí)
97  GO
98
99  EXECUTE master.dbo.xp_delete_file
100     0,                                    -- Tipo de archivo (0=Backup)
101     @DirectorioBackup,                   -- Directorio de backups de QhatuPERU
102     N'trn',                               -- Extensión para Logs
103     @FechaCorte,
104     1;
105  GO
106
107
```

Justificación Técnica

La ejecución automatizada de un script de limpieza es crucial para la **gestión del espacio en disco**. La **Política de Retención (7 días)** establece el máximo periodo al que la empresa puede retroceder para recuperar datos. El procedimiento **xp_delete_file** permite borrar archivos en el sistema operativo basándose en el tipo de archivo, la extensión y la marca de tiempo, garantizando que solo los respaldos fuera de la política sean eliminados.

Buenas Prácticas

- Permisos y Seguridad:** Si usa **xp_cmdshell** o **xp_delete_file**, asegúrese de deshabilitar **xp_cmdshell** después de su uso o usar un Proxy Account con permisos mínimos si se ejecuta como Job.
- Pruebas:** Pruebe el script de limpieza manualmente antes de automatizarlo para confirmar que solo borra los archivos deseados.
- Notificación:** Configure el trabajo de limpieza en el Agente SQL Server para que envíe una **alerta** si el proceso falla, indicando un posible problema de permisos o de espacio.

6. Restauración punto en el tiempo

Script T-SQL

```
110  -- 1. Asumir Full Backup a las 00:00:00 y Logs cada 15 min. Falla a las 09:30:00.
111  -- Restaurar el Full Backup
112 RESTORE DATABASE [QhatuPERU_PIT]
113   FROM DISK = 'D:\SQLBackups\QhatuPERU_FULL.bak'
114   WITH NORECOVERY, REPLACE;
115   GO
116
117  -- 2. Restaurar Logs intermedios con NORECOVERY hasta el Log que contiene la hora de falla
118 RESTORE LOG [QhatuPERU_PIT]
119   FROM DISK = 'D:\SQLBackups\QhatuPERU_LOG_0900.trn' -- Log que contiene el rango de 09:00:00 a 09:15:00
120   WITH NORECOVERY;
121   GO
122
123  -- 3. Restaurar el Log CRÍTICO, deteniéndose justo antes del error (09:29:00)
124 RESTORE LOG [QhatuPERU_PIT]
125   FROM DISK = 'D:\SQLBackups\QhatuPERU_LOG_0915.trn' -- Log que contiene el rango de 09:15:00 a 09:30:00
126   WITH **STOPAT** = '2025-11-20 09:29:00', -- Detiene las transacciones en este punto
127     ~RECOVERY; -- Pone la BD en línea en el estado 09:29:00
128   GO
129
130
131
```

Justificación Técnica

La cláusula **WITH STOPAT** es la herramienta que permite lograr la restauración a un punto en el tiempo (PIT). Esta cláusula le indica al motor de SQL Server que, durante el proceso de *roll forward* (aplicación de transacciones), se detenga exactamente en la fecha y hora especificadas, asegurando que las transacciones posteriores (incluyendo el error lógico) no sean aplicadas. Esto **minimiza la pérdida de datos** a un minuto o menos.

Buenas Prácticas

- **Precisión de STOPAT:** Sea lo más preciso posible al definir la hora. Si tiene dudas, deténgase un poco antes y valide los datos.
- **Restauración en Staging:** Siempre restaure a una base de datos temporal (QhatuPERU_PIT) primero. Esto le permite **validar** que la tabla **ARTICULO** se recuperó correctamente antes de aplicarlo al entorno de producción.
- **Modelo Full Obligatorio:** Recuerde que esta funcionalidad **solo es posible** si la base de datos QhatuPERU está en el modelo de recuperación **Full**.