

LISTING 2-1 Multiple backups to four backup devices

```
-- Initial full backup
BACKUP DATABASE AdventureWorks TO
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice1.bak'
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice2.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 3.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 4.bak '
WITH
FORMAT'
MEDIANAME = ' AdventureWorksMediaSet1';
GO

-- Differential backup
BACKUP DATABASE AdventureWorks TO
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice1.bak'
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 2.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 3.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 4.bak '
WITH
NOINIT'
MEDIANAME = 'AdventureWorksMediaSet1'
DIFFERENTIAL;
GO

-- Incremental backup
BACKUP LOG AdventureWorks TO
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice1.bak'
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 2.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 3.bak '
DISK = 'R:\SQLBackup\AdventureWorks_BackupDevice 4.bak '
WITH
NOINIT'
MEDIANAME = 'AdventureWorksMediaSet1';
GO
```

LISTING 2-2 Mirrored backups

```
BACKUP DATABASE WorldWideImporters
TO DISK = 'B:\SQLBackup\WorldWideImporters.bak'
MIRROR TO DISK = '\\DEVSERVER\SQLBackup\SQLBackup\WorldWideImporters.bak'
MIRROR TO DISK = '\\TESTSERVER\SQLBackup\SQLBackup\WorldWideImporters.bak'
MIRROR TO DISK = '\\STAGINGSERVER\SQLBackup\SQLBackup\WorldWideImporters.bak'
WITH FORMAT;
```

LISTING 2-3 Progress of backup operation

```
SELECT session_id' db_name(database_id) as database_name'
       start_time' command' percent_complete' estimated_completion_time
FROM sys.dm_exec_requests
WHERE command LIKE 'backup %';
```

LISTING 2-4 Create a database snapshot

```
CREATE DATABASE WorldWideImporters_20160917
ON (
    NAME = WorldWideImporters_Data'
    FILENAME = 'R:\SQLData\WorldWideImporters_20160917.ss')
AS SNAPSHOT OF [WorldWideImporters];
```

LISTING 2-5 Backup to URL

```
-- Create storage account identity and access key
CREATE CREDENTIAL MyCredentialName
WITH IDENTITY = 'MyStorageAccountName',
SECRET = '<MyStorageAccountAccessKey>';
GO
-- Backup database to URL using storage account identity and access key
BACKUP DATABASE MyDB
TO URL = 'https://<MyStorageAccountName>.blob.core.windows.
net/<MyStorageAccountContainerName>/MyDB.bak'
WITH CREDENTIAL = 'MyCredentialName'
```

LISTING 2-6 SQL Server Managed Backup to Microsoft Azure at the instance level

```
USE [msdb];
GO
EXEC [smart_admin].[sp_set_instance_backup]
    @enable_backup=1
    ,@storage_url = 'https://mystorageaccount.blob.core.windows.net/'
    ,@retention_days=30
    ,@credential_name='MyCredential'
    ,@encryption_algorithm = 'AES_256'
    ,@encryptor_type= 'ServerCertificate'
    ,@encryptor_name='MyBackupCertificate';
GO
```

LISTING 2-7 SQL Server Managed Backup to Microsoft Azure at the instance level

```
USE [msdb];
GO
EXEC [smart_admin].[sp_set_db_backup]
    @database_name='MyDB'
    ,@enable_backup=1
    ,@storage_url = 'https://MyStorageAccount.blob.core.windows.net/'
    ,@retention_days=30
    ,@credential_name='MyCredential'
    ,@encryption_algorithm = 'NO_ENCRYPTION';
GO
```

LISTING 2-8 Creating a VLDB

```
CREATE DATABASE [VLDB]
    ON [PRIMARY]
        (NAME = N'VLDB_System' FILENAME = N'D:\SQLData\VLDB_System.mdf' SIZE = 100MB)
    FILEGROUP [DATA]
        (NAME = N'VLDB_Data1' FILENAME = N'E:\SQLData\VLDB_Data1.ndf' SIZE = 1TB)
        (NAME = N'VLDB_Data2' FILENAME = N'F:\SQLData\VLDB_Data2.ndf' SIZE = 1TB)
        (NAME = N'VLDB_Data3' FILENAME = N'G:\SQLData\VLDB_Data3.ndf' SIZE = 1TB)
        (NAME = N'VLDB_Data4' FILENAME = N'H:\SQLData\VLDB_Data4.ndf' SIZE = 1TB)
        (NAME = N'VLDB_Data5' FILENAME = N'I:\SQLData\VLDB_Data5.ndf' SIZE = 1TB)
    LOG ON
        (NAME = N'VLDB_log' FILENAME = N'L:\SQLLog\VLDB_Log.ldf' SIZE = 100GB) ;
GO
ALTER DATABASE [VLDB] MODIFY FILEGROUP [DATA] DEFAULT;
```

LISTING 2-9 Query the recovery model of all databases

```
USE master;
GO
SELECT name, recovery_model_desc
FROM sys.databases;
```

LISTING 2-10 Orphaned log experiment

```
-- Set up experiment: Run selectively as and if required
/* You might have to enable xp_cmdshell by running the following:
EXEC sp_configure 'show advanced' 1;
RECONFIGURE;
GO
EXEC sp_configure 'xp_cmdshell' 1;
RECONFIGURE;
GO
-- Create directory for experiment
EXEC xp_cmdshell 'md C:\Exam764Ch2\';
GO
*/
-- Create database
CREATE DATABASE TailLogExperimentDB
ON PRIMARY (NAME = N'TailLogExperimentDB_data' FILENAME = N'C:\Exam764Ch2\
TailLogExperimentDB.mdf')
LOG ON (NAME = N'TailLogExperimentDB_log' FILENAME = N'C:\Exam764Ch2\
TailLogExperimentDB.ldf')
GO
-- Create table
USE [TailLogExperimentDB]
GO
CREATE TABLE [MyTable] (Payload VARCHAR(1000));
GO
-- Insert first record
INSERT [MyTable] VALUES ('Before full backup');
GO
-- Perform full backup
BACKUP DATABASE [TailLogExperimentDB] TO DISK = 'C:\Exam764Ch2\TailLogExperimentDB_FULL.
bak' WITH INIT;
GO
-- Insert second record
INSERT [MyTable] VALUES ('Before log backup');
GO
-- Perform log backup
BACKUP LOG [TailLogExperimentDB] TO DISK = 'C:\Exam764Ch2\TailLogExperimentDB_LOG.bak'
WITH INIT;
GO
-- Insert third record
INSERT [MyTable] VALUES ('After log backup');
GO
-- Simulate disaster
SHUTDOWN;
/*
Perform the following actions:
    1. Use Windows Explorer to delete C:\Exam764Ch2\TailLogExperimentDB.mdf
    2. Use SQL Server Configuration Manager to start SQL Server
The [TailLogExperimentDB] database should now be damaged as you deleted the primary data
file.
*/
```

LISTING 2-11 Attempted log backup

```
USE master;
SELECT name, state_desc FROM sys.databases WHERE name = 'TailLogExperimentDB';
GO
-- Try to back up the orphaned tail-log
BACKUP LOG [TailLogExperimentDB] TO DISK = 'C:\Exam764Ch2\TailLogExperimentDB_
OrphanedLog.bak' WITH INIT;
```

LISTING 2-12 Orphaned log backup with NO_TRUNCATE option

```
-- Try to back up the orphaned tail-log again
BACKUP LOG [TailLogExperimentDB] TO DISK = 'C:\Exam764Ch2\TailLogExperimentDB_
OrphanedLog.bak' WITH NO_TRUNCATE;
```

LISTING 2-13 Orphaned log experiment cleanup

```
-- Cleanup experiment: Run selectively as and if required
/*
EXEC xp_cmdshell 'rd /q C:\Exam764Ch2\';
GO
EXEC sp_configure 'xp_cmdshell' 0;
RECONFIGURE;
GO
EXEC sp_configure 'show advanced' 0;
RECONFIGURE;
GO
USE [master];
DROP DATABASE [TailLogExperimentDB];
*/
```

LISTING 2-14 Querying the database's transaction log reuse wait

```
USE master;
GO
SELECT [database_id], [name] as 'database_name', [state_desc], [recovery_mode]_desc],
[log_reuse_wait_desc]
FROM [sys].[databases];
```

LISTING 2-15 KILL statement

```
KILL 69;
```

LISTING 2-16 Scheduling a backup through SQL Server Agent

```
USE [msdb ]
GO
/***** Object: Job [Full Backup - System Databases]    Script Date: 26/09/2017
11:56:45 PM *****/
BEGIN TRANSACTION
DECLARE @ReturnCode INT
SELECT @ReturnCode = 0
/***** Object: JobCategory [DBA]    Script Date: 26/09/2017 11:56:45 PM *****/
IF NOT EXISTS (SELECT name FROM msdb.dbo.syscategories WHERE name=N'DBA' AND category_
class=1)
BEGIN
EXEC @ReturnCode = msdb.dbo.sp_add_category @class=N'JOB'' @type=N'LOCAL'' @name=N'DBA'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
END

DECLARE @jobId BINARY(16)
EXEC @ReturnCode =  msdb.dbo.sp_add_job @job_name=N'Full Backup - System Databases''
        @enabled=1'
        @notify_level_eventlog=0'
        @notify_level_email=2'
        @notify_level_netsend=0'
        @notify_level_page=0'
        @delete_level=0'
        @description=N'Perform a full database backup of the following system databases:
- master
- msdb

Schedule:
- Nightly at 2:00AM''
        @category_name=N'DBA''
        @owner_login_name=N'sa''
        @notify_email_operator_name=N'Victor Isakov'' @job_id = @jobId OUTPUT
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [Backup master database]    Script Date: 26/09/2017 11:56:45 PM
*****/
EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId' @step_name=N'Backup master
database''
        @step_id=1'
        @cmdexec_success_code=0'
        @on_success_action=3'
        @on_success_step_id=0'
        @on_fail_action=3'
        @on_fail_step_id=0'
        @retry_attempts=3'
        @retry_interval=5'
        @os_run_priority=0' @subsystem=N'TSQL''
        @command=N'BACKUP DATABASE master
TO DISK = ''R:\SQLBackup\master.bak''
WITH INIT;'
        @database_name=N'master''
        @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
/***** Object: Step [Backup msdb database]    Script Date: 26/09/2017 11:56:45 PM
*****/
```

```

EXEC @ReturnCode = msdb.dbo.sp_add_jobstep @job_id=@jobId' @step_name=N'Backup msdb
database''
    @step_id=2'
    @cmdexec_success_code=0'
    @on_success_action=1'
    @on_success_step_id=0'
    @on_fail_action=2'
    @on_fail_step_id=0'
    @retry_attempts=0'
    @retry_interval=0'
    @os_run_priority=0' @subsystem=N'TSQL''
    @command=N'BACKUP DATABASE msdb
TO DISK = 'R:\SQLBackup\msdb.bak''
WITH INIT;''
    @database_name=N'master''
    @flags=0
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_update_job @job_id = @jobId' @start_step_id = 1
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobschedule @job_id=@jobId' @name=N'Nighty @ 02:00''
    @enabled=1'
    @freq_type=4'
    @freq_interval=1'
    @freq_subday_type=1'
    @freq_subday_interval=0'
    @freq_relative_interval=0'
    @freq_recurrence_factor=0'
    @active_start_date=20170926'
    @active_end_date=99991231'
    @active_start_time=20000'
    @active_end_time=235959'
    @schedule_uid=N'47e87563-0e09-4856-821d-ee181839d09'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
EXEC @ReturnCode = msdb.dbo.sp_add_jobserver @job_id = @jobId' @server_name = N'(local)'
IF (@@ERROR <> 0 OR @ReturnCode <> 0) GOTO QuitWithRollback
COMMIT TRANSACTION
GOTO EndSave
QuitWithRollback:
    IF (@@TRANCOUNT > 0) ROLLBACK TRANSACTION
EndSave:

```

LISTING 2-17 Create [Backup Error] alert

```
USE [msdb]
GO
EXEC msdb.dbo.sp_add_alert @name=N'Backup Error'
    @message_id=3041'
    @severity=0'
    @enabled=1'
    @delay_between_responses=300'
    @include_event_description_in=1'
    @category_name=N'[Uncategorized]''
    @job_id=N'00000000-0000-0000-0000-000000000000'
```

LISTING 2-18 Verifying a backup set

```
RESTORE VERIFYONLY
FROM DISK = 'B:\SQLBackup\WorldWideImporters_FULL.bak'
```

LISTING 2-19 Restoring a database

```
USE [master];
GO
-- Restore full backup
RESTORE DATABASE [WorldWideImporters]
    FROM DISK = '\\SQLBACKUPS\SQLBackup\WorldWideImporters\WorldWideImporters_FULL.bak'
    WITH NORECOVERY'
    MOVE 'WorldWideImporters_Data' TO 'D:\SQLData\WorldWideImporters_Data.mdf'
    MOVE 'WorldWideImporters_Log' to 'L:\SQLLog\WorldWideImporters_Log.ldf';
GO
-- Restore differential backup
RESTORE DATABASE [WorldWideImporters]
    FROM DISK = '\\SQLBACKUPS\SQLBackup\WorldWideImporters\WorldWideImporters_DIFF.bak'
    WITH NORECOVERY;
GO
-- Restore last 3 log backups
RESTORE LOG [WorldWideImporters]
    FROM DISK = '\\SQLBACKUPS\SQLBackup\WorldWideImporters\WorldWideImporters_LOG.bak'
    WITH FILE = 11' NORECOVERY;
GO
RESTORE LOG [WorldWideImporters]
    FROM DISK = '\\SQLBACKUPS\SQLBackup\WorldWideImporters\WorldWideImporters_LOG.bak'
    WITH FILE = 12' NORECOVERY;
GO
RESTORE LOG [WorldWideImporters]
    FROM DISK = '\\SQLBACKUPS\SQLBackup\WorldWideImporters\WorldWideImporters_LOG.bak'
    WITH FILE = 13' NORECOVERY;
GO
-- Make database available
RESTORE DATABASE [WorldWideImporters]
    WITH RECOVERY;
GO
```


LISTING 2-20 Restoring last log backup

```
RESTORE LOG [WorldWideImporters]
FROM DISK = '\\SQLBACKUPS\SQLBackup\WordWideImporters\WordWideImporters_LOG.bak'
WITH FILE = 13' RECOVERY;
```

LISTING 2-21 Revert a Database Snapshot

```
USE master;
GO
RESTORE DATABASE [WorldWideImporters]
FROM DATABASE_SNAPSHOT = 'WorldWideImporters_20160917';
```

LISTING 2-22 [OnlineStore] database definition

```
CREATE DATABASE [OnlineStore]
ON PRIMARY
( NAME = N'OnlineStore_Primary', FILENAME = N'D:\SQLData\OnlineStore_Primary.mdf'
, SIZE = 100MB),
FILEGROUP [Archive]
( NAME = N'OnlineStore_Archive', FILENAME = N'E:\SQLData\OnlineStore_Archive.ndf'
, SIZE = 1TB),
FILEGROUP [CompletedOrders]
( NAME = N'OnlineStore_CompletedOrders', FILENAME = N'F:\SQLData\OnlineStore_
CompletedOrders.ndf' , SIZE = 200GB),
FILEGROUP [Data]
( NAME = N'OnlineStore_Data', FILENAME = N'G:\SQLData\OnlineStore_Data.ndf' , SIZE
= 10GB),
FILEGROUP [Orders]
( NAME = N'OnlineStore_Orders', FILENAME = N'H:\SQLData\OnlineStore_Orders.ndf' ,
SIZE = 20GB)
LOG ON
( NAME = N'OnlineStore_Log', FILENAME = N'L:\SQLLog\OnlineStore_Log.1df' , SIZE =
1GB)
GO
ALTER DATABASE [OnlineStore]
MODIFY FILEGROUP [Data] DEFAULT;
```

LISTING 2-23 Partial-restore sequence

```
-- Back up orphaned transaction log to minimize data loss
USE [master];
GO
BACKUP LOG [OnlineStore] TO DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.bak' WITH
NO_TRUNCATE;
-- Start partial-restore sequence
USE [master]
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'PRIMARY' FROM DISK = 'B:\SQLBackup\OnlineStore_FULL.bak' WITH
NORECOVERY, PARTIAL;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'PRIMARY' FROM DISK = 'B:\SQLBackup\OnlineStore_DIFF.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'PRIMARY' FROM DISK = 'B:\SQLBackup\OnlineStore_LOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'PRIMARY' FROM DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore] WITH RECOVERY;
```

LISTING 2-24 [Orders] filegroup-restore sequence

```
USE [master];
GO
-- Restore Orders filegroup and bring it online
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Orders' FROM DISK = 'B:\SQLBackup\OnlineStore_FULL.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Orders' FROM DISK = 'B:\SQLBackup\OnlineStore_DIFF.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Orders' FROM DISK = 'B:\SQLBackup\OnlineStore_LOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Orders' FROM DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore] WITH RECOVERY
```

LISTING 2-25 Check partial availability of database files

```
USE [OnlineStore];
GO
-- Check to see if [Orders] filegroup is online
SELECT file_id, name, type_desc, state_desc
FROM sys.database_files
GO
-- Ensure users can query the critical tables
SELECT * FROM [Orders]
```

LISTING 2-26 [Data] and [CompletedOrders] filegroup-restore sequence

```
USE [master];
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Data', FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\
OnlineStore_FULL.bak' WITH NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Data', FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\
OnlineStore_DIFF.bak' WITH NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Data', FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\
OnlineStore_LOG.bak' WITH NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Data', FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\
OnlineStore_ORPHANEDLOG.bak' WITH NORECOVERY;
GO
RESTORE DATABASE [OnlineStore] WITH RECOVERY
GO
```

LISTING 2-27 [Archive] filegroup-restore sequence

```
USE [master];
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Archive' FROM DISK = 'B:\SQLBackup\OnlineStore_FULL.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Archive' FROM DISK = 'B:\SQLBackup\OnlineStore_DIFF.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Archive' FROM DISK = 'B:\SQLBackup\OnlineStore_LOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'Archive' FROM DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore] WITH RECOVERY
GO
```

LISTING 2-28 [CompletedOrders] filegroup-restore sequence for a partially available database

```
USE [master];
GO
-- Take the [CompletedOrders] file offline
ALTER DATABASE
MODIFY FILE (name = OnlineStore_CompletedOrders' OFFLINE);
GO
-- Back up orphaned transaction log to minimize data loss
BACKUP LOG [OnlineStore] TO DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.bak' WITH
NO_TRUNCATE;
GO
--
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\OnlineStore_FULL.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\OnlineStore_DIFF.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\OnlineStore_LOG.bak' WITH
NORECOVERY;
GO
RESTORE DATABASE [OnlineStore]
FILEGROUP = 'CompletedOrders' FROM DISK = 'B:\SQLBackup\OnlineStore_ORPHANEDLOG.
bak' WITH NORECOVERY;
GO
RESTORE DATABASE [OnlineStore] WITH RECOVERY
GO
```

LISTING 2-29 Page restore

```
USE [msdb];
GO
-- Determine corrupted pages
SELECT database_id' file_id' page_id' event_type' error_count' last_update_date
FROM dbo.suspect_pages
WHERE database_id = DB_ID('WorldWideImporters');
GO
-- Restore 4 corrupt pages
USE [master];
GO
RESTORE DATABASE [WorldWideImporters] PAGE='1:300984' 1:300811' 1: 280113' 1:170916'
FROM WideWorldImporters_Backup_Device
WITH NORECOVERY;
GO
RESTORE LOG [WorldWideImporters]
FROM WideWorldImporters_Log_Backup_Device
WITH FILE = 1' NORECOVERY;
GO
RESTORE LOG [WorldWideImporters]
FROM WideWorldImporters_Log_Backup_Device
WITH FILE = 2' NORECOVERY;
WITH NORECOVERY;
GO
BACKUP LOG [WorldWideImporters]
TO DISK='B:\SQLBackup\PageRecovery.bak';
GO
RESTORE LOG <database>
FROM DISK='B:\SQLBackup\PageRecovery.bak'
WITH RECOVERY;
GO
```

LISTING 2-30 Point-in-time recovery with a date/time

```
RESTORE LOG WideWorldImporters
FROM WideWorldImporters_Log_Backup_Device
WITH FILE=6' RECOVERY'
STOPAT = 'Sep 17' 2016 2:30 AM';
```

LISTING 2-31 Restoring a database to a LSN

```
-- Set up experiment
/* You might have to enable xp_cmdshell by running the following:
EXEC sp_configure 'show advanced' 1;
RECONFIGURE;
GO
EXEC sp_configure 'xp_cmdshell' 1;
RECONFIGURE;
GO
*/
USE [master];
EXEC xp_cmdshell 'md C:\Exam764Ch2\';
GO

-- Create database
CREATE DATABASE [RestoreToLSNExperimentDB]
ON PRIMARY (
    NAME = N'RestoreToLSNExperiment_data'
    FILENAME = N'C:\Exam764Ch2\RestoreToLSNExperiment.mdf')
LOG ON (
    NAME = N'RestoreToLSNExperiment_log'
    FILENAME = N'C:\Exam764Ch2\RestoreToLSNExperiment.ldf')
GO
USE [RestoreToLSNExperimentDB]
GO

-- Create table
CREATE TABLE [MyTable] (Payload VARCHAR(1000))
GO

-- Step 1: Insert 3 records
INSERT [MyTable] VALUES ('Record 1') ('Record 2') ('Record 3')
GO

-- Step 2: Perform full backup
BACKUP DATABASE [RestoreToLSNExperimentDB]
TO DISK = 'C:\Exam764Ch2\RestoreToLSNExperimentDB_FULL.bak'
WITH INIT;
GO

-- Step 3: Insert 1 record
INSERT [MyTable] VALUES ('Record 4');
GO

SELECT * FROM [MyTable]
-- Step 4: Query the transaction log
SELECT * FROM fn_dblog(NULL,NULL);
-- Record the last LSN: 0x00000025:000000a9:0007
GO

-- Step 5: Accidentally update all 4 records (simulating a mistake)
UPDATE [MyTable] SET Payload = 'MISTAKE'
GO

SELECT * FROM [MyTable]
SELECT * FROM fn_dblog(NULL,NULL);
-- Perform log backup
BACKUP LOG [RestoreToLSNExperimentDB]
TO DISK = 'C:\Exam764Ch2\RestoreToLSN_LOG.bak'
WITH INIT;
GO

-- Step 6: Drop database
USE master;
DROP DATABASE [RestoreToLSNExperimentDB];
-- Step 7: Restore full backup
RESTORE DATABASE [RestoreToLSNExperimentDB]
FROM DISK = 'C:\Exam764Ch2\RestoreToLSNExperimentDB_FULL.bak'
```

```

WITH NORECOVERY;
-- Step 8: Restore log backup at LSN recorded above
RESTORE LOG [RestoreToLSNExperimentDB]
FROM DISK = 'C:\Exam764Ch2\RestoreToLSNExperimentDB_LOG.bak'
WITH RECOVERY' STOPATMARK = 'lsn:0x00000025:000000a9:0007';
-- Step 9: Confirm restore doesn't include "mistake"
USE [RestoreToLSNExperimentDB];
SELECT * FROM [MyTable];
-- Cleanup experiment
/*
USE master;
DROP DATABASE [RestoreToLSNExperimentDB];
GO
EXEC xp_cmdshell 'rd /s /q C:\Exam764Ch2;
GO
EXEC sp_configure 'xp_cmdshell' 0;
RECONFIGURE;
GO
EXEC sp_configure 'show advanced' 0;
RECONFIGURE;
GO
*/

```

LISTING 2-32 Create a transaction log mark

```

USE [WideWorldImporters];
GO
BEGIN TRANSACTION PriceIncrease
    WITH MARK 'PriceIncrease';
UPDATE [Warehouse].[StockItems]
    SET [RecommendedRetailPrice] = [RecommendedRetailPrice] * 1.25;
COMMIT TRANSACTION PriceIncrease;
GO

```

LISTING 2-33 Query all marked transactions

```

SELECT [database_name]' [mark_name]' [description]' [user_name]' [lsn]' [mark_time]
FROM [msdb].[dbo].[logmarkhistory];

```

LISTING 2-34 Restore database to marked transaction

```

USE [WorldWideImporters];
GO
RESTORE LOG [WideWorldImporters]
FROM DISK = 'B:\SQLBackup\WideWorldImporters_LOG.bak'
WITH RECOVERY'
STOPATMARK = 'PriceIncrease';
GO

```


LISTING 2-35 Filegroup restore example

```
USE master;
GO
-- Restore filegroup
RESTORE DATABASE [WordWideImporters]
    FILEGROUP = 'WordWideImporters_OrderHistory'
    FROM WordWideImporters_Backup_OrderHistory_Device
    WITH NORECOVERY'
    REPLACE;
GO
-- Restore first log backup
RESTORE LOG [WordWideImporters]
    FROM WordWideImporters_Log_Device
    WITH FILE = 1' NORECOVERY;
GO
-- Restore second log backup and recover database
RESTORE LOG [WordWideImporters]
    FROM WordWideImporters_Log_Device
    WITH FILE = 2' RECOVERY;
GO
```

LISTING 2-36 DBCC operation progress

```
SELECT session_id' db_name(database_id) as database_name'
        start_time' command' percent_complete' estimated_completion_time
FROM sys.dm_exec_requests
WHERE command LIKE 'dbcc%';
```

LISTING 2-37 DBCC consistency check

```
DBCC CHECKDB ([AdventureWorksDW])
WITH EXTENDED_LOGICAL_CHECKS' TABLOCK' NO_INFOMSGS;
GO
```

LISTING 2-38 Repairing a database using the REPAIR_REBUILD operation

```
USE [master];
-- Change database to single user mode
ALTER DATABASE [WideWorldImporters] SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO
-- Perform safe repair
DBCC CHECKDB ([WideWorldImporters]) WITH REPAIR_REBUILD;
GO
```

LISTING 2-39 Repairing a database using the REPAIR_ALLOW_DATA_LOSS operation

```
USE [master];
GO
ALTER DATABASE [WideWorldImporters] SET EMERGENCY;
GO
ALTER DATABASE [WideWorldImporters] SET SINGLE_USER;
GO
DBCC CHECKDB ('WideWorldImporters' REPAIR_ALLOW_DATA_LOSS)
WITH NO_INFOMSGS;
GO
```