Microsoft

SQL Server Disaster Recovery

Module 7

Learning Units covered in this Module

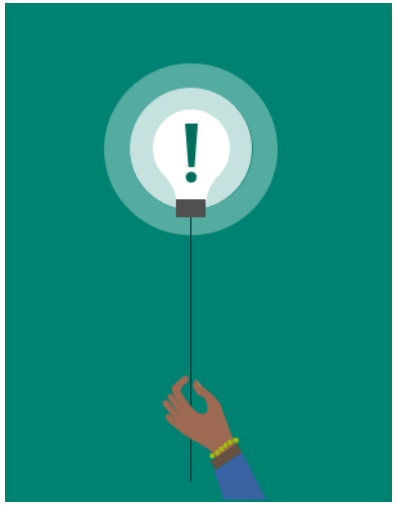
- Lesson 1: Basic Disaster Recovery
- Lesson 2: Backups
- Lesson 3: Restores
- Lesson 4: Backup Features

Lesson 1: Basic Disaster Recovery

Objectives

After completing this learning, you will be able to:

- Understand how the different types of backups and restores fit into a recovery strategy.
- Determine which recovery models are necessary for proper backups and restores.
- Use SLAs to create a recovery strategy.



Disaster Recovery

Covers scenarios where data loss is at risk:

Types of Backups

Database Backups

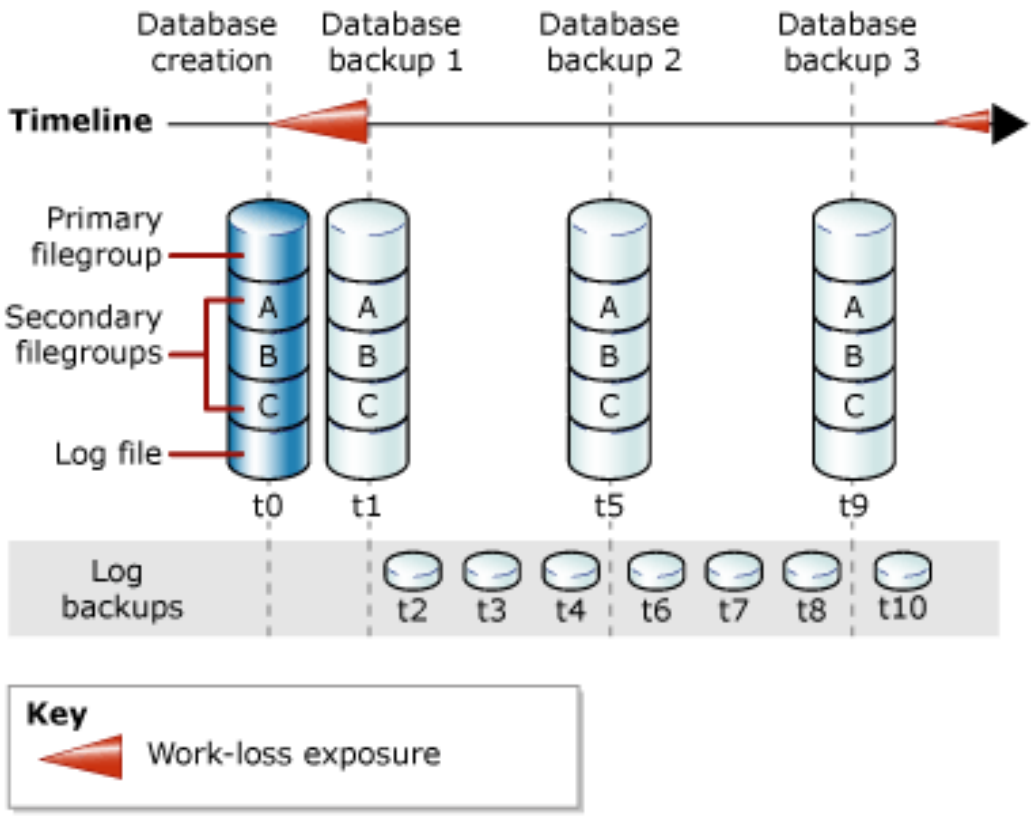
Full Backup | Differential Backup

Transaction Log Backup

Full Log backup | Tail log backup

Partial Backups

File and Filegroup



Database Recovery Models

Simple

- Automatic truncation of the log

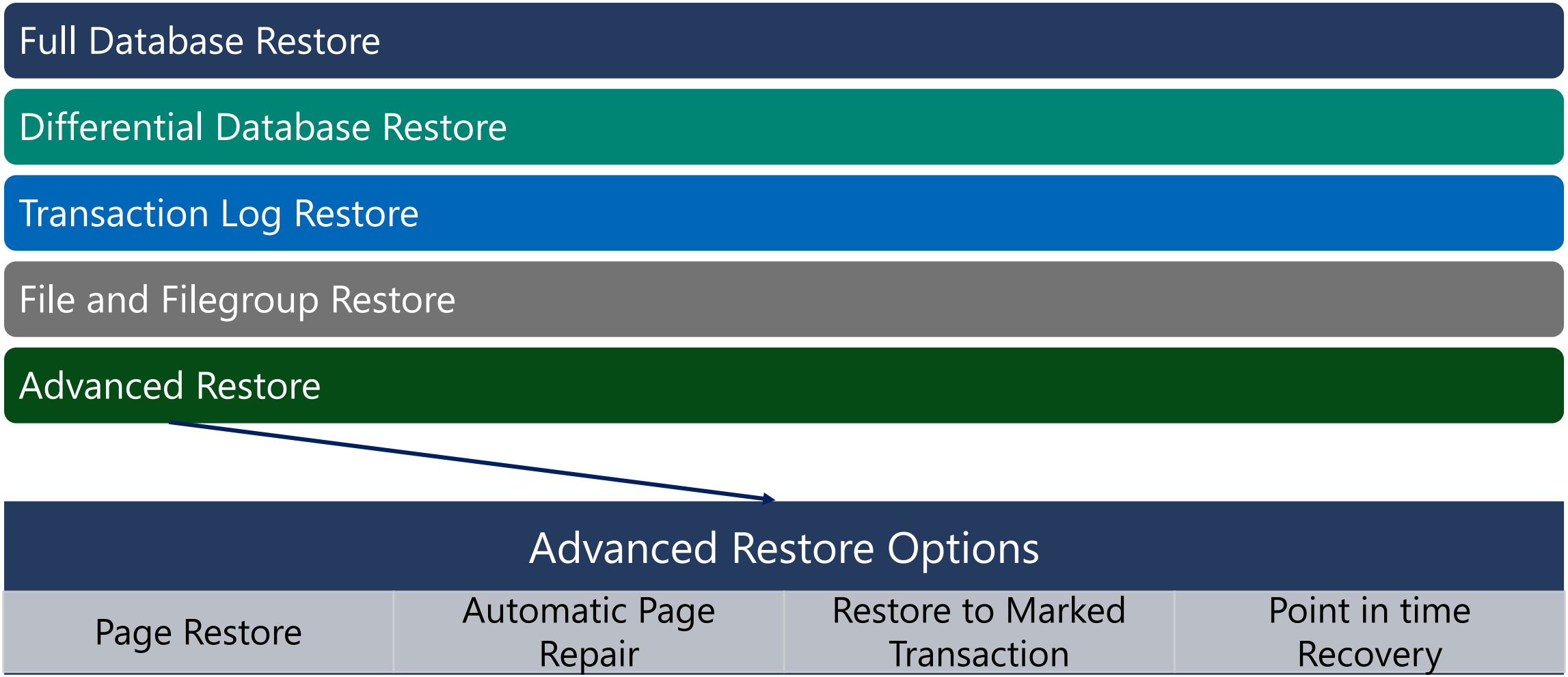
Full

- Requires transaction log backups to manage file growth
- All operations are fully logged
- Point in time recovery permitted

Bulk Logged

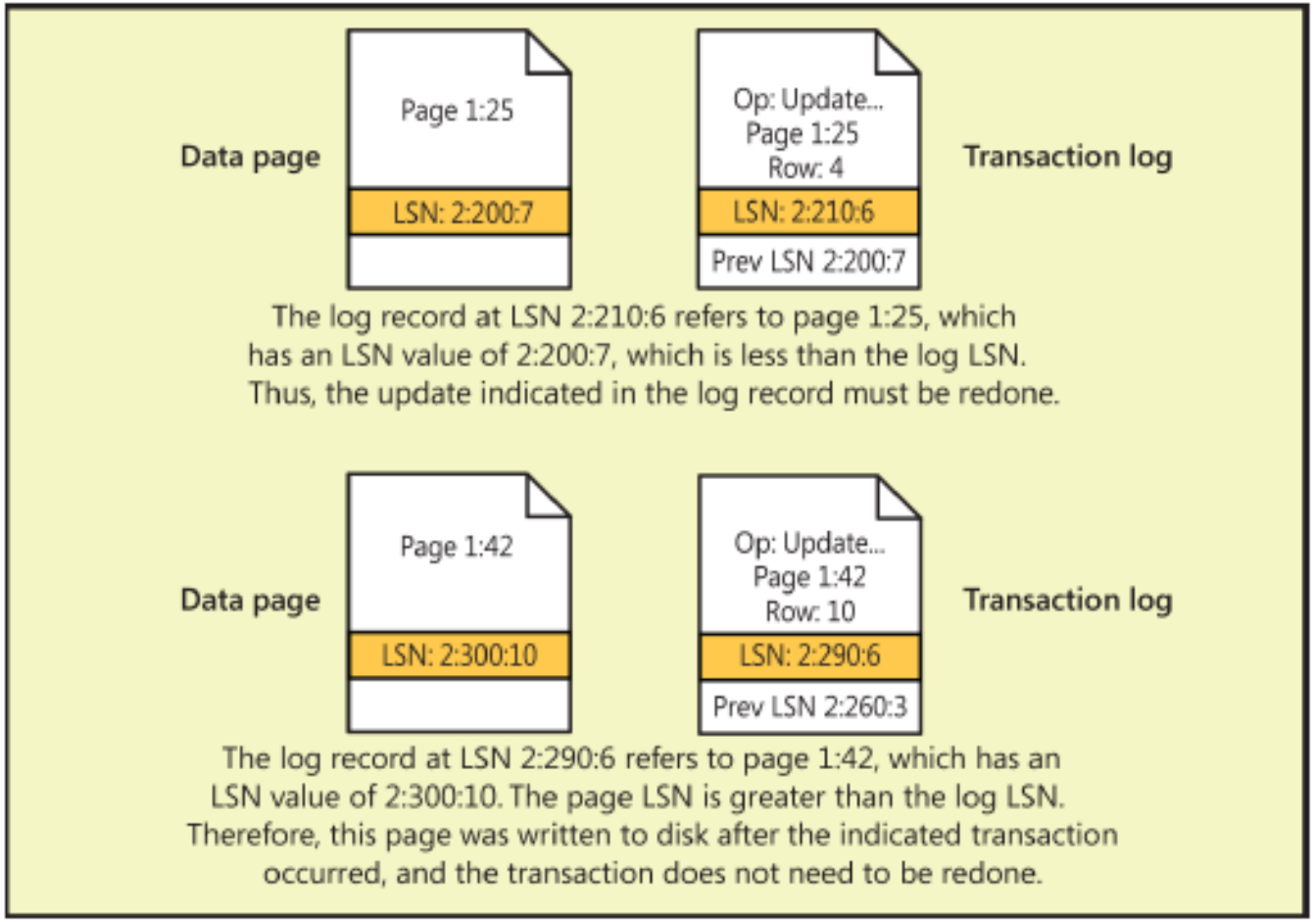
- Certain bulk operations such as BULK INSERT, SELECT INTO, INDEX REBUILD and other are minimally logged

Types of Restores

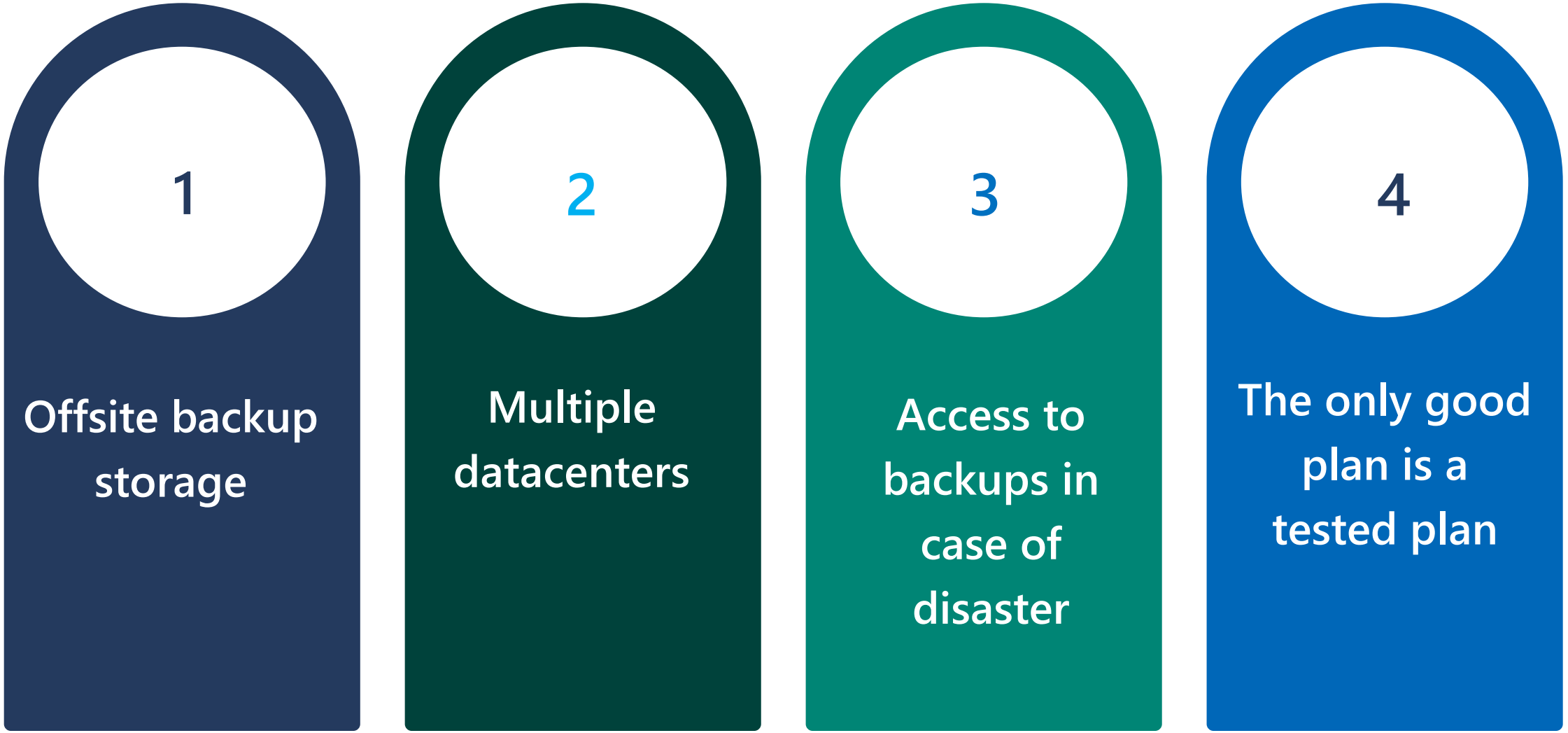


Page LSN and recovery

- Last LSN in the page header
- Log Record has two LSNs
  - Previous LSN from the data page
  - Current log record LSN
- All used for recovery



Can my plan survive a true disaster?



Building a Restore Strategy

Scenario 1

An application does a nightly load of data from secondary sources into your database. Throughout the day, users only report off of this data.		
What are your ideas on how you would restore this database if necessary?	Which recovery model would you choose?	What additional information could be useful in defining a recovery plan?

Building a Restore Strategy

Scenario 2

An application does a nightly load of data from secondary sources into your database. Throughout the day, users report off the data but also update the data.	
How does your strategy change from Scenario 1?	Again, what extra information is important to be able to define an accurate plan?

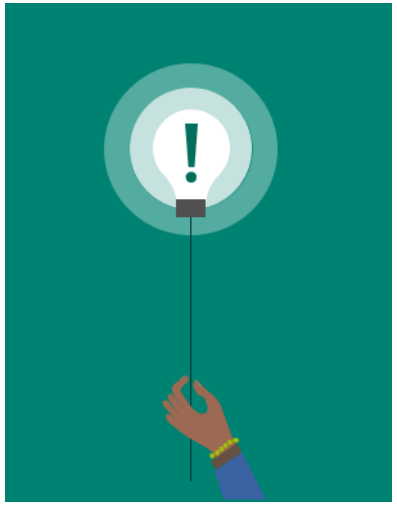
Questions?

Lesson 2: Backups

Objectives

After completing this learning, you will be able to:

- Identify which types of backups are suitable for different recovery plans.
- Explain how the choice of backup affects storage and performance.



Full Backup

Backup of the entire database

Only allocated pages are included in the backup. Free space in database files is not.

Includes data up to the point in time when the reading portion of the backup is completed.

Does not allow recovery to a specific point in time.

Always includes some transaction log data.

Allows the backup to be transactionally consistent even though Data Manipulation Language (DML) statements are allowed during a backup operation.

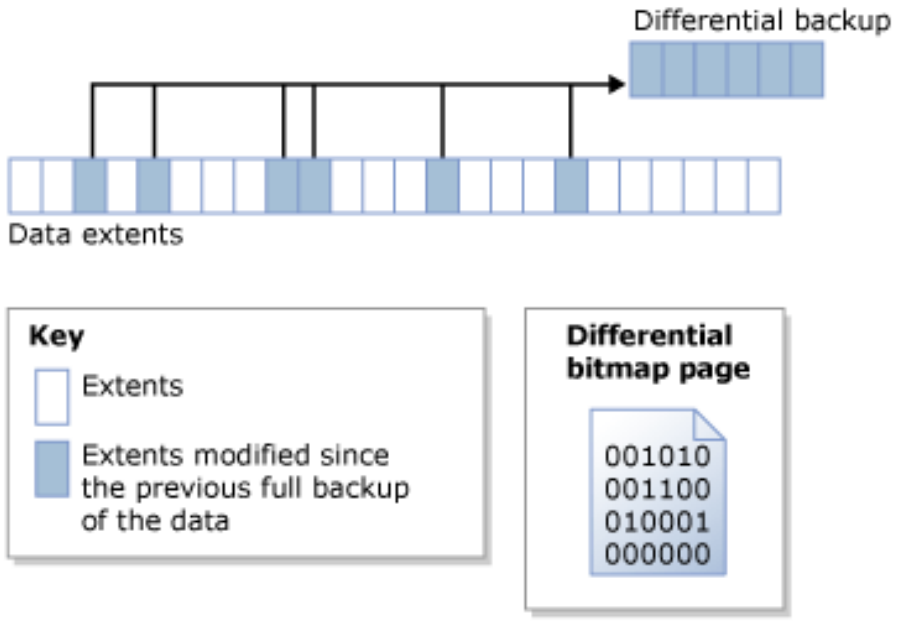
Differential Backups

Backup of the extents changed since the last full database backup

Store active part of the transaction log

Faster to restore than restoring transaction log backups for the same time period

Independent of other differential backups



Discussion

Differential Backups

How do differential backups fit into a DR plan?

Advantages over Full backups?

How often to do differential backups?

Incremental vs differential?

Transaction Log Backups

Includes log records since the last log backup (incremental backups)

Inactive log records are attempted to be truncated afterwards (may not always occur)

Database must be in full or bulk-logged recovery model

Required for point-in-time recovery

Discussion

Transaction Log Backups	
How often should you take transaction log backups?	How does having transaction log backups effect your restore plan?

File/Filegroup Backups

Provides functionality to backup a single file or a single filegroup

Useful for large databases

Allows for flexible backup and restore

Limited use in Simple Recovery Model

Discussion

File and Filegroup Backups	
How can filegroup backups be utilized to meet your recovery time objective (RTO)?	How can filegroup backups be used to reduce overall backup storage?

Demonstration

Backing Up Databases

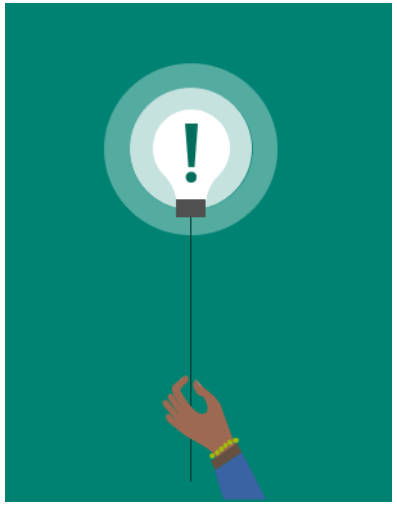
Questions?

Lesson 3: Restores

Objectives

After completing this learning, you will be able to:

- Identify which types of restores are needed for different recovery plans.
- Explain how restores impact RTO/RPO.



Restores and Service Level Agreements

Restore is dependent on how the database was backed up

Given the backups, can the database be restored to meet RTO and RPO objectives?

Simple Strategies:

- Use full backups and transaction log backups
- Generally can meet RPO objectives
- As databases increase in size, RTO can be difficult to achieve

Complex Strategies

- Use of Differential, File, FileGroup backups
- Piecemeal Restore
- Can reduce time to restore
- Supporting Documentation needed

Full and Differential Database Restores

- Both Full and Differential use the RESTORE DATABASE command
- Use WITH DIFFERENTIAL for differential restores
- Full backups must be restored first
- Only the LAST differential database backup needs to be restored
- Use WITH REPLACE to overwrite an existing database
- Use WITH NORECOVERY to leave the database in a restoring state to apply addition backups
- The last restore command should use WITH RECOVERY to bring the database online

File/Filegroup Backups and Piecemeal Restores

1

Large database can be restored in pieces

2

Primary filegroup must always be restored

3

Transaction log restores needed to bring all filegroups to the same point in time.

4

Online filegroup restores are enterprise edition only

Questions?

Knowledge Check

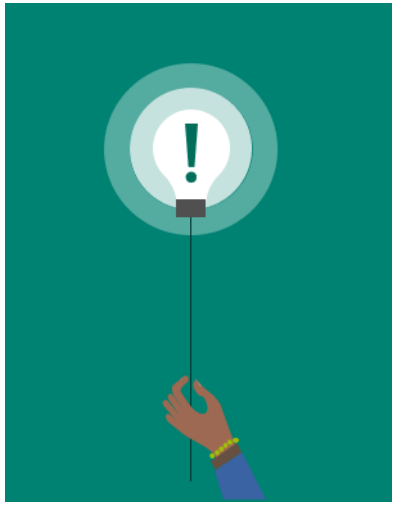
It is Friday morning at 8:00 AM. What is the order of operations to restore a database that has a full backup on Monday at 2:00 AM, a differential backup every other weeknight at 2:00 AM, and transaction log backups every 15 minutes? The database must be restored to 6:00 AM Friday morning.

Lesson 4: Backup Features

Objectives

After completing this learning, you will be able to:

- Understand the features used with backup.



Accelerated Database Recovery (ADR)

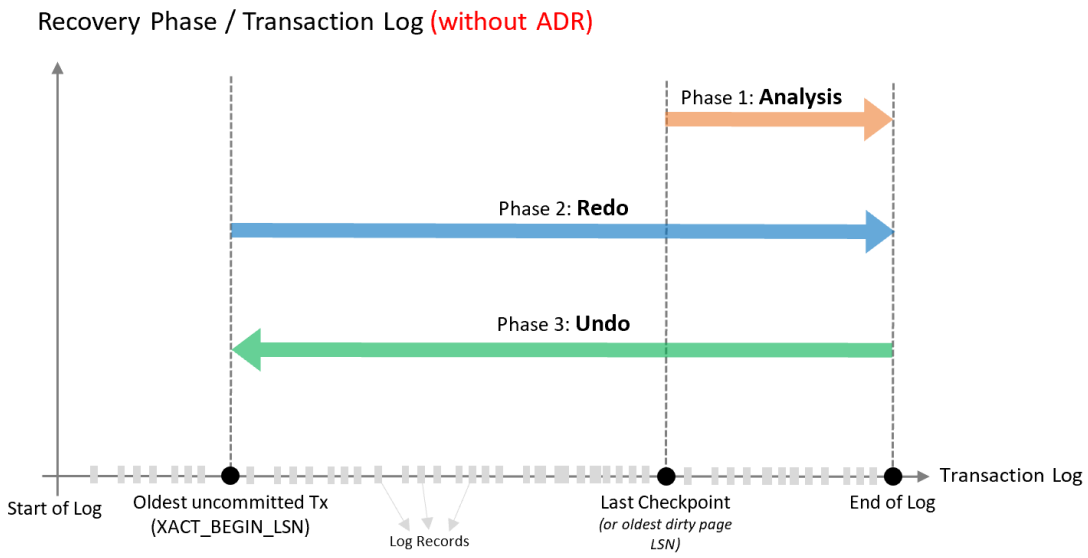
- Redesign of the database recovery process starting in SQL Server 2019
- Improve database availability especially when long transactions are present

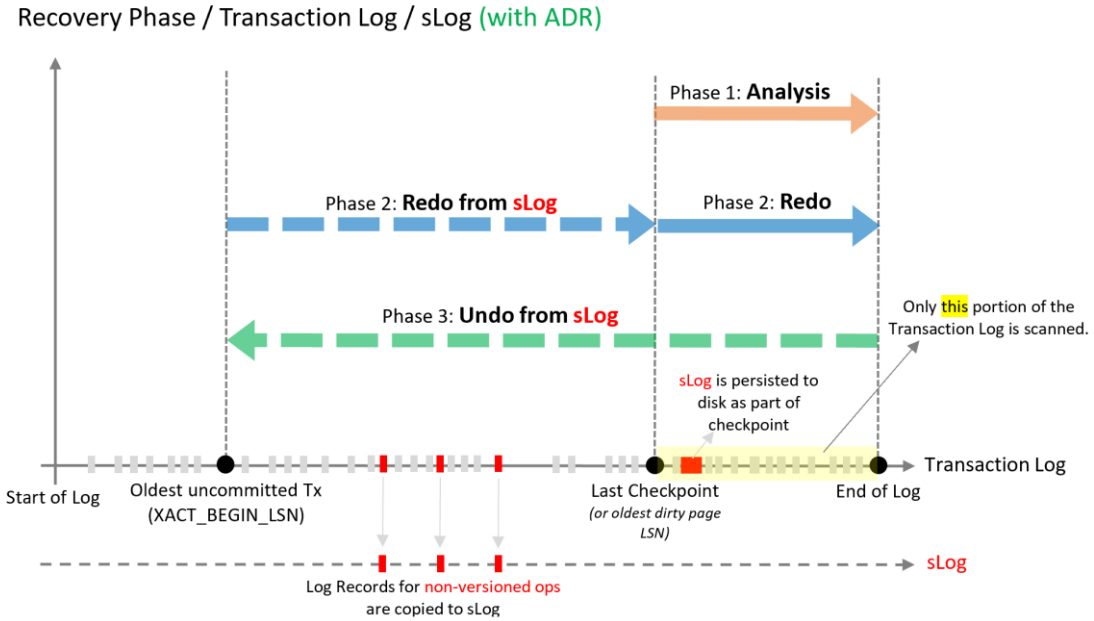
Instantaneous transaction rollback

Fast Consistent database recovery

Aggressive log truncation

ADR vs current recovery





Accelerated Database Recovery

Components

Persisted Version Store (PVS)

- Version Store containing database changes stored in the database itself

Logical Revert

- Keeps track of all aborted transactions
- Performs rollback using PVS

sLog

- Secondary in-memory log stream
- Persisted on-disk by checkpoint process
- Processes non-versioned operations

Cleaner

- Purges page versions no longer needed

Demonstration

Accelerated Database Recovery

Encrypted Backups

Supports on-premises and Azure Blob Storage

Configurable for Managed Backup to Microsoft Azure

Starting in SQL Server 2016 backup, encryption is now supported with compression by using AES-NI hardware acceleration

Backup Encryption Best Practices

Back up all keys

- Service master key
- Database master key
- Certificate used for the backup

Offsite backup of certificates

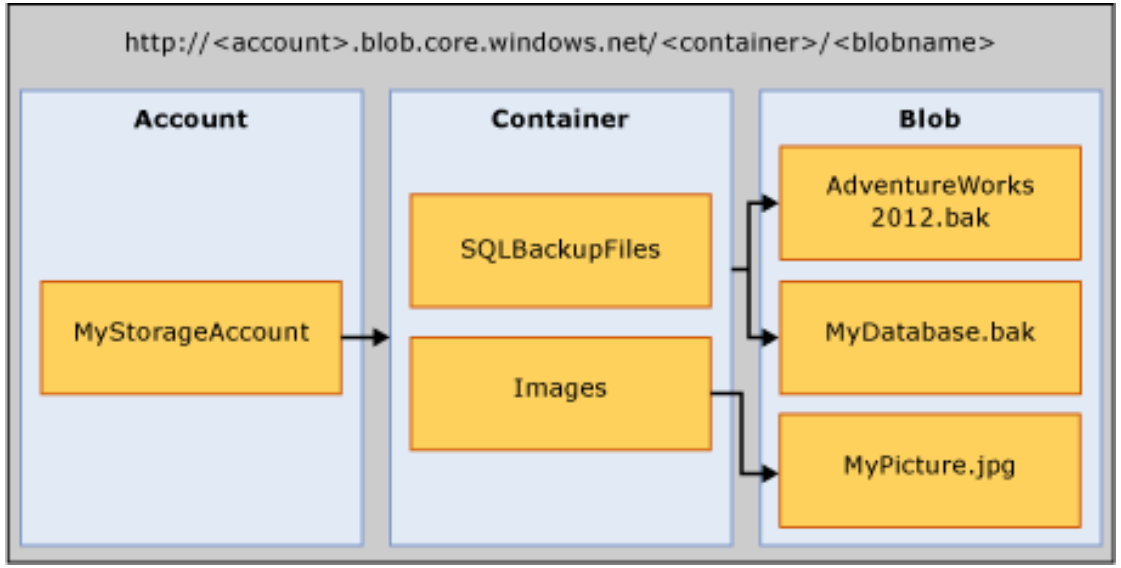
Do not store the certificate backup and the encrypted backup in the same location

Do not expire or renew certificates

Monitor the CPU impact of encrypting backups

Backup To Azure Blobs

- Two times cheaper storage (than Backup to Azure in prior versions)
- Backup striping and faster restore
- Maximum backup size is over 12 terabytes (TB)
- Granular access and unified credential story (Shared Access Signature (SAS) Uniform Resource Identifiers (URIs)
- Supports all existing backup/restore features (except append)



Backup To Azure Blocks

Utilizes the BACKUP TO URL functionality

Two key components:

- Storage Account
- URL Path

Supports compression and encryption

```
CREATE CREDENTIAL [https://<account>.blob.core.windows.net/<container>]
WITH IDENTITY = 'Shared Access Signature',
SECRET = 'sig=mw3K6dpwV%2BWUPj8L4Dq3cyNxCI'

BACKUP DATABASE database TO
 URL = N'https://<account>.blob.core.windows.net/<container>/<blob1>',
 URL = N'https://<account>.blob.core.windows.net/<container>/<blob2>'
```

Managed Backup to Azure

Managed Backup to Microsoft Azure uses new block blob storage for backup files

Stripe backup sets, enabling backup file sizes up to 12.8 TB

Other changes and enhancements to managed backup:

- Managed Backup used for system databases
- Support for databases in full, bulk logged, and simple recovery model
- Support for both automated and custom scheduling of backups
- Customized backup schedules – full backup and log backup

Demonstration

Backup to Azure

Questions?

Knowledge Check

What are some of the key benefits of Accelerated Database Recovery?

Can backup compression and encryption be combined?

When encrypting backups, what other item also needs to be backed up?