Working with Databases and Storage

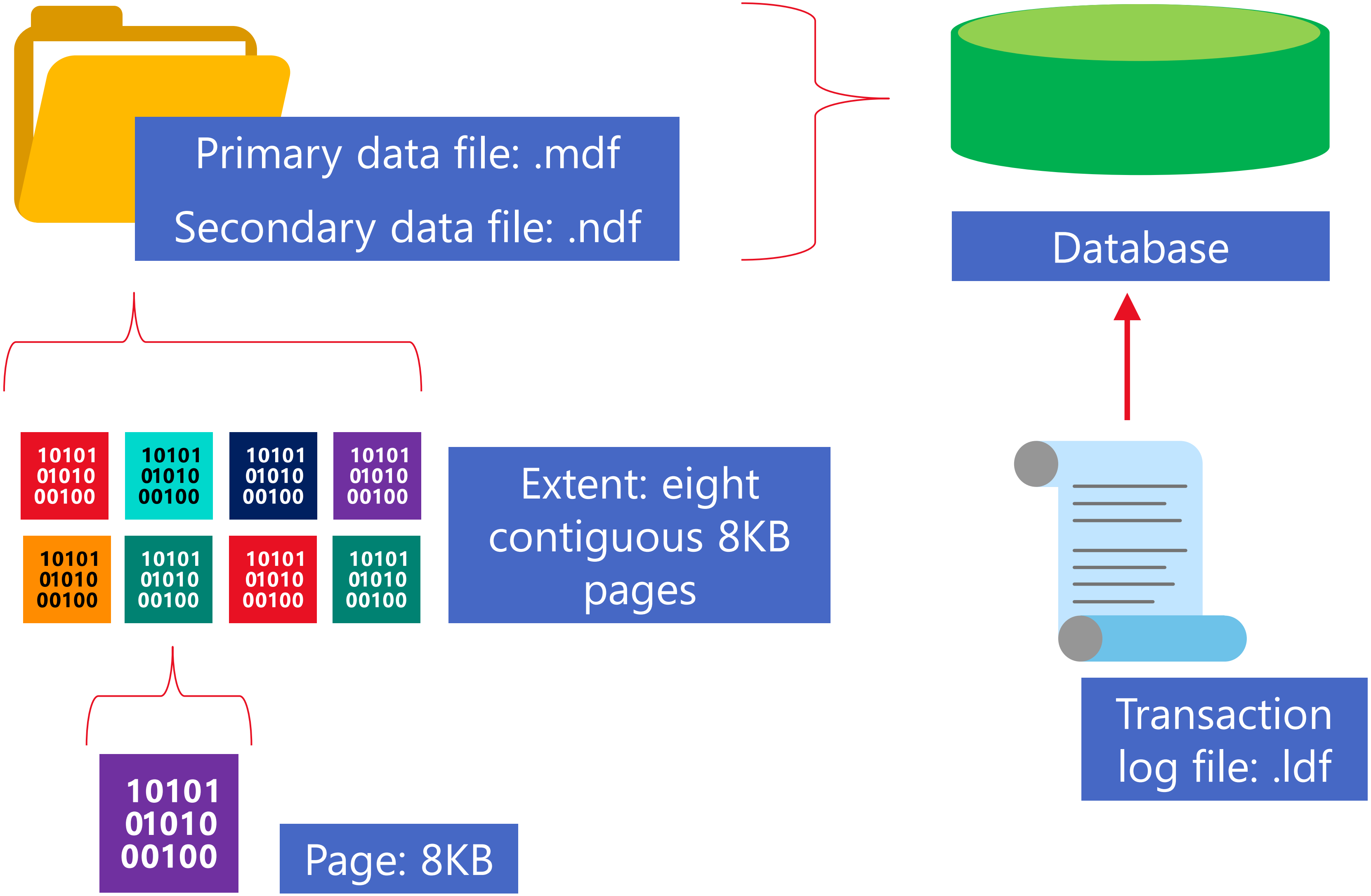
Module Overview

- Introduction to Data Storage with SQL Server
- Managing Storage for System Databases
- Managing Storage for User Databases
- Moving Database Files
- Configuring the Buffer Pool Extension

Data Storage with SQL Server

- How Data Is Stored in SQL Server
- Considerations for Disk Storage Devices
- Determining File Placement and Number of Files
- Ensuring Sufficient File Capacity

How Data Is Stored in SQL Server



Primary data file: .mdf

Secondary data file: .ndf

Database

Extent: eight contiguous 8KB pages

Page: 8KB

Transaction log file: .ldf

Considerations for Disk Storage Devices

- Direct Attached Storage—disks connected by a RAID controller
- Storage Area Network—disks connected by a network and available to multiple servers
- Windows Storage Pools—commodity disk drives grouped together to create one large storage space

File Placement and Number of Files

- Isolate log and data files at the physical disk level
- Determine the number and location of data files based on performance and maintenance considerations
  - Use additional files to spread data across storage locations
  - Use smaller data files when easier maintenance is needed
  - Use data files as units of backup and restore
- Determine log file requirements
  - Use a single log file in most situations as log files are written sequentially

Ensuring Sufficient File Capacity

- Estimate the size of data, log files and tempdb
  - Perform load testing with the actual application
  - Check with the application vendor

- Set the size to a reasonable size:
  - Leave enough space for new data, without the need to regularly expand
  - Monitor data and log file usage
  - Plan for manual expansion
  - Keep autogrowth enabled to allow for unexpected growth

Managing Storage for System Databases

- SQL Server System Databases
- Moving System Databases
- Considerations for tempdb
- Demonstration: Moving tempdb Files

SQL Server System Databases

System Database	Description
master	Stores all system-level configuration
msdb	Holds SQL Server Agent configuration data
model	Provides the template for new databases
tempdb	Holds temporary data
resource	Contains system objects that are mapped to the sys schema of databases

Moving System Databases

- Moving **msdb** and **model**, and **tempdb**
  1. Execute ALTER DATABASE…MODIFY FILE for each file
  2. Stop the SQL Server service
  3. Move the files
  4. Restart the SQL Server service
- Moving **master**
  1. Change the **–d** and **–l** startup parameters for the SQL Server service
  2. Stop the SQL Server service
  3. Manually move the files while the instance is stopped
  4. Restart the SQL Server service

Misconfiguration can prevent SQL Server from starting

Considerations for tempdb

- **tempdb**:
  - Contains temporary data for internal objects, row versioning, and user objects
  - Is truncated or rebuilt with every restart of the instance
  - Occupies varying amounts of space
  - Should be tested with real-life workloads

- Place **tempdb** on a fast and separate I/O subsystem to ensure good performance:
- Split **tempdb** into data files of equal size per core

Demonstration: Moving tempdb Files

In this demonstration, you will see how to:

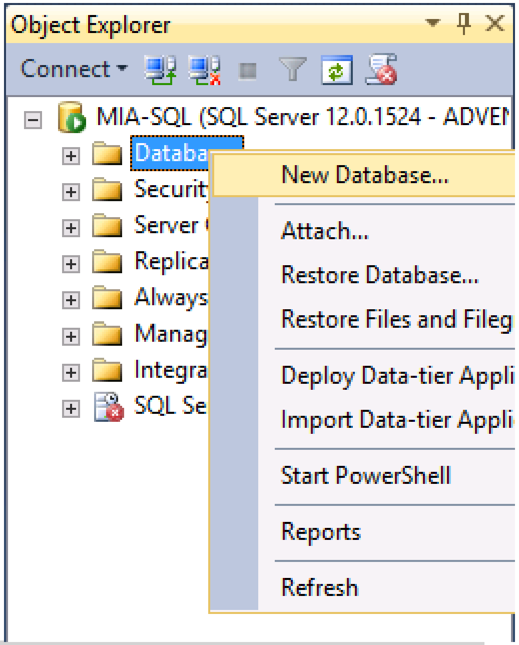
- Move **tempdb** files

Managing Storage for User Databases

- Creating User Databases
- Configuring Database Options
- Demonstration: Creating Databases
- Altering User Databases
- Managing Database Files
- Introduction to Filegroups
- Creating and Managing Filegroups

Creating User Databases

- Create databases:
  - In SQL Server Management Studio
  - By using the CREATE DATABASE statement



```
CREATE DATABASE Sales
ON
  (NAME = Sales_dat, FILENAME = ‘M:\Data\Sales.mdf',
   SIZE = 100MB, MAXSIZE = 500MB, FILEGROWTH = 20% )
LOG ON
  (NAME = Sales_log, FILENAME = 'L:\Logs\Sales.ldf',
   SIZE = 20MB, MAXSIZE = UNLIMITED, FILEGROWTH = 10MB );
```

Configuring Database Options

- Database-level options are unique to each database

Option	Description
Auto options	Define whether some operations should occur automatically within the database
Page verify	Define how the page should be verified when read from disk; should be set to CHECKSUM
Recovery model	Defines the recovery model of the database
State options	Sets the state of the database, such as Online/Offline, Restricted Access or Read Only

Demonstration: Creating Databases

In this demonstration, you will see how to:

- Create a database by using SQL Server Management Studio
- Create a database by using the CREATE DATABASE statement

Altering User Databases

- Altering database options:

```
ALTER DATABASE HistoricSales
SET READ_ONLY;
```

- Altering database compatibility:

```
ALTER DATABASE Sales
SET COMPATIBILITY_LEVEL = 100;
```

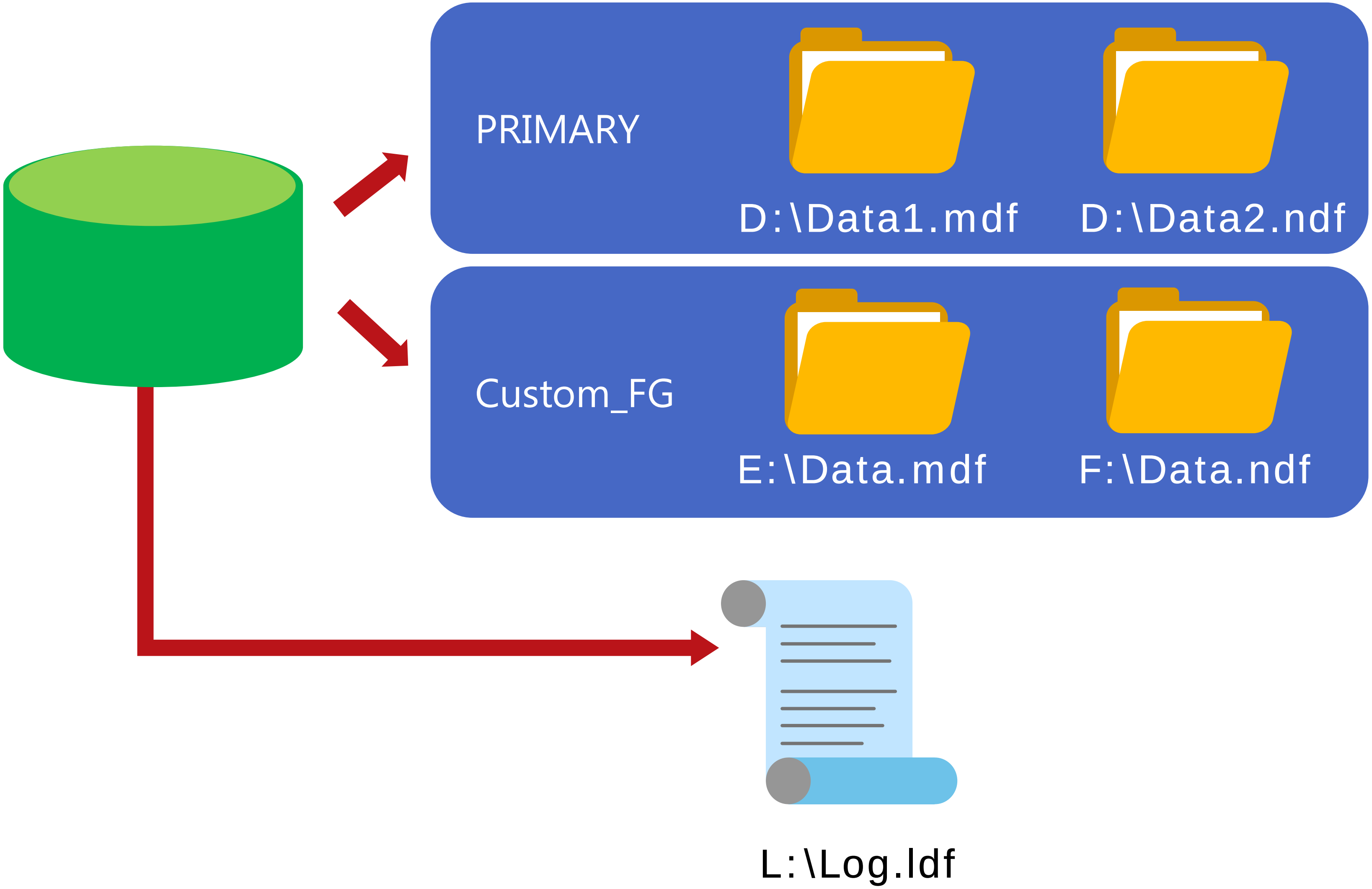
Managing Database Files

- Adding space to a database:
  - ALTER DATABASE … ADD FILE
  - ALTER DATABASE … MAXSIZE

- Dropping database files:
  1. Empty file: DBCC SHRINKFILE … EMPTYFILE
  2. Drop file: ALTER DATABASE

- Shrinking databases:
  - DBCC SHRINKDATABASE
  - DBCC SHRINKFILE
  - TRUNCATE_ONLY

Introduction to Filegroups



Creating and Managing Filegroups

- Creating filegroups
  - CREATE DATABASE … FILEGROUP (<*files*>)
  - ALTER DATABASE … ADD FILEGROUP <*filegroup*>

- Setting the default filegroup
  - ALTER DATABASE … MODIFY FILEGROUP <*filegroup*> DEFAULT

- Using read-only filegroups
  - ALTER DATABASE … MODIFY FILEGROUP <filegroup> READONLY

Lesson 4: Moving Database Files

- Moving User Database Files
- Detaching and Attaching Databases
- Demonstration: Detaching and Attaching a Database

Moving User Database Files

- Data and log files can be moved within the instance:
  - Database must be offline

- ALTER DATABASE statement:
  - For copying within an instance
  - Manually move files on the file system

Detaching and Attaching Databases

- Detaching a database unhooks the database from the instance:
  - Data and log files are kept intact
  - Detached files can be attached again on the same or a different instance

- Use detach/attach to move databases to other instances

- Detach/attach is useful in disaster recovery situations

Demonstration: Detaching and Attaching a Database

In this demonstration, you will see how to:

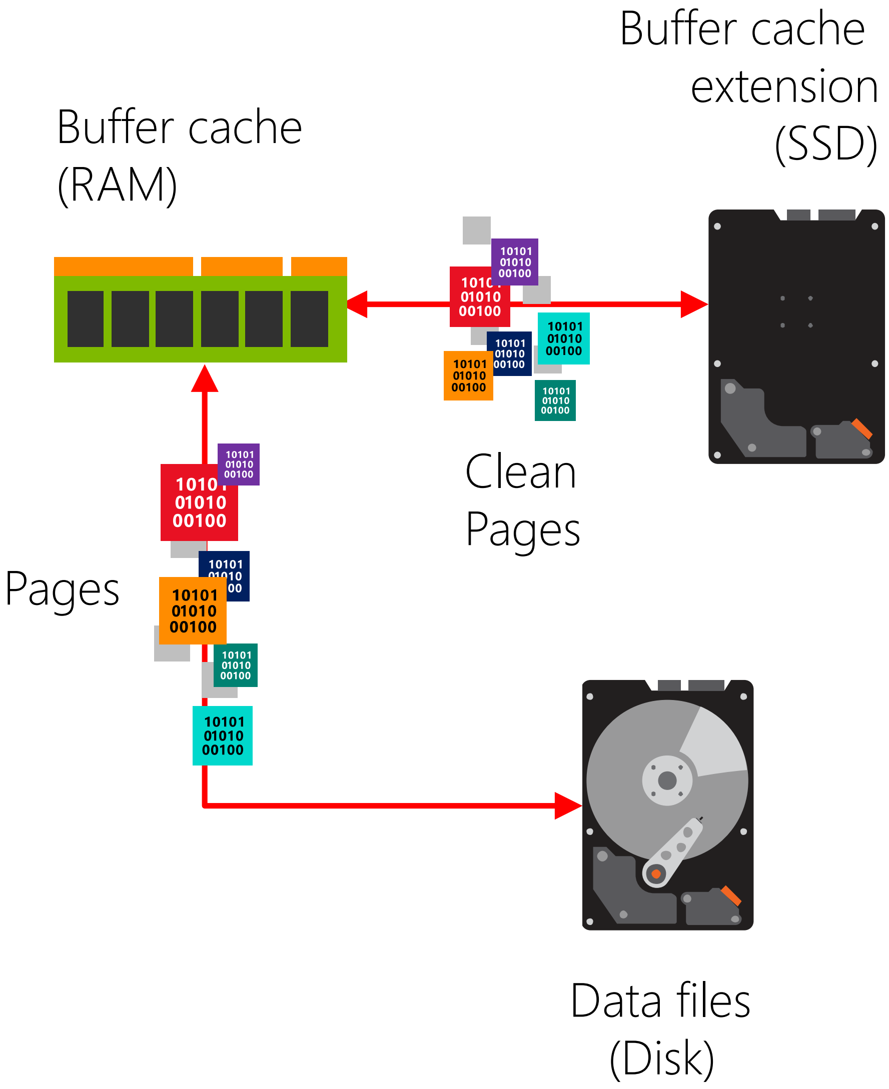
- Detach a database
- Attach a database

Lesson 5: Configuring the Buffer Pool Extension

- Introduction to the Buffer Pool Extension
- Considerations for Using the Buffer Pool Extension
- Configuring the Buffer Pool Extension
- Demonstration: Configuring the Buffer Pool Extension

Introduction to the Buffer Pool Extension

- Extends buffer cache to non-volatile storage
- Improves performance for read-heavy OLTP workloads
- SSD devices are often more cost effective than adding physical memory
- Simple configuration with no changes to existing applications



Considerations for Using the Buffer Pool Extension

- Improves performance for OLTP databases where:
  - OLTP operations with a high volume of reads
  - Up to 32 GB of physical memory
  - Buffer Pool Extension is 4x to 10x physical memory
  - Buffer Pool Extension on high throughput SSD storage

- Unlikely to improve performance for:
  - Data warehouse workloads
  - OLTP workloads with a high volume of write operations
  - Servers with more than 64 GB of physical memory for SQL Server

Configuring the Buffer Pool Extension

- Enable using ALTER SERVER CONFIGURATION

```
ALTER SERVER CONFIGURATION
SET BUFFER POOL EXTENSION ON
(FILENAME = 'E:\SSDCACHE\MYCACHE.BPE',
 SIZE = 50 GB);
```

- To Reconfigure, disable and then re-enable

Demonstration: Configuring the Buffer Pool Extension

In this demonstration, you will see how to:

- Enable the Buffer Pool Extension
- Verify Buffer Pool Extension Configuration
- Disable the Buffer Pool Extension

SQL Server DBA Exercises-1

Core Database Management

1. Add 10mb to MDDB data file, 3 MB to the log file write the TSQL Commands

2. Add 20 MB to NORTHWIND DB data file. Restrict it to 100 MB. Write the TSQL command

3. Create TESTDB Database. Add a secondary file with 10 MB size.
Shrink the database 30%. Write all the TSQL Commands.

4. Move TEMPDB to c:\temp\db directory. Write the TSQL commands
Explain the process

5-Change page verify to torn page detection for NORTHWIND. Write the TSQL commands.

6- Detach and attach the NORTHWIND database to new directory c:\SQL\DATA. Write the TSQL Command

7-Add a file group called IMGFG to NORTHWIND.
Enter a 10 MB size NDF file to it. Write the TSQL Commands