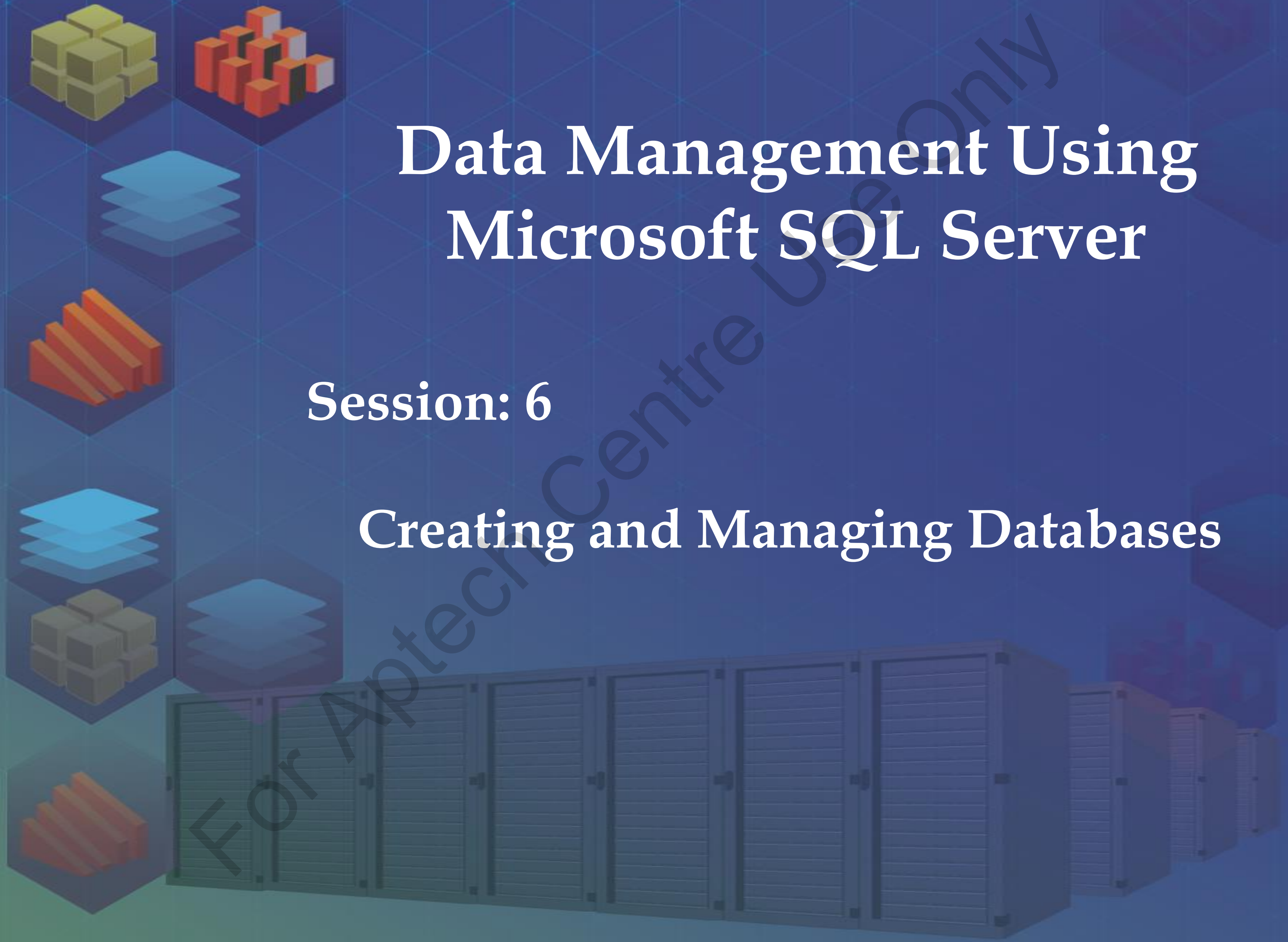


Data Management Using Microsoft SQL Server

Session: 6

Creating and Managing Databases

SQL Server 2012

Objectives

- Describe system and user-defined databases
- List the key features of the AdventureWorks2012 sample database
- Describe adding of filegroups and transaction logs
- Describe the procedure to create a database
- List and describe types of database modifications
- Describe the procedure to drop a database
- Describe database snapshots

SQL Server 2012

Introduction 1-3

- A database is a collection of data stored in data files on a disk or some removable medium.
- A database consists of data files to hold actual data.
- An SQL Server database is made up of a collection of tables that stores sets of specific structured data.
- A table includes a set of rows (also called as records or tuples) and columns (also called as attributes).
- Each column in the table is intended to store a specific type of information, for example, dates, names, currency amounts, and numbers.
- A user can install multiple instances of SQL Server on a computer.
- Each instance of SQL Server can include multiple databases.
- Within a database, there are various object ownership groups called schemas.

Introduction 2-3

- Within each schema, there are database objects such as tables, views, and stored procedures.
- Some objects such as certificates and asymmetric keys are contained within the database, but are not contained within a schema.
- SQL Server databases are stored as files in the file system.
- These files are grouped into file groups.
- When people gain access to an instance of SQL Server, they are identified as a login.
- When people gain access to a database, they are identified as a database user.
- A user who has access to a database can be given permission to access the objects in the database.

Introduction 3-3

- Though permissions can be granted to individual users, it is recommended to create database roles, add the database users to the roles, and then, grant access permission to the roles.
- Granting permissions to roles instead of users makes it easier to keep permissions consistent and understandable as the number of users grow and continually change.
- SQL Server 2012 supports three kinds of databases, which are as follows:

System Databases

User-defined Databases

Sample Databases

SQL Server 2012

System Databases

- SQL Server uses system databases to support different parts of the DBMS.
- Each database has a specific role and stores job information that requires to be carried out by SQL Server.
- The system databases store data in tables, which contain the views, stored procedures, and other database objects.
- They also have associated database files (for example, `.mdf` and `.ldf` files) that are physically located on the SQL Server machine.
- Following table shows the system databases that are supported by SQL Server 2012:

Database	Description
master	The database records all system-level information of an instance of SQL Server.
msdb	The database is used by SQL Server Agent for scheduling database alerts and various jobs.
model	The database is used as the template for all databases to be created on the particular instance of SQL Server 2012.
resource	The database is a read-only database. It contains system objects included with SQL Server 2012.
tempdb	The database holds temporary objects or intermediate result sets.

SQL Server 2012

Modifying System Data 1-3

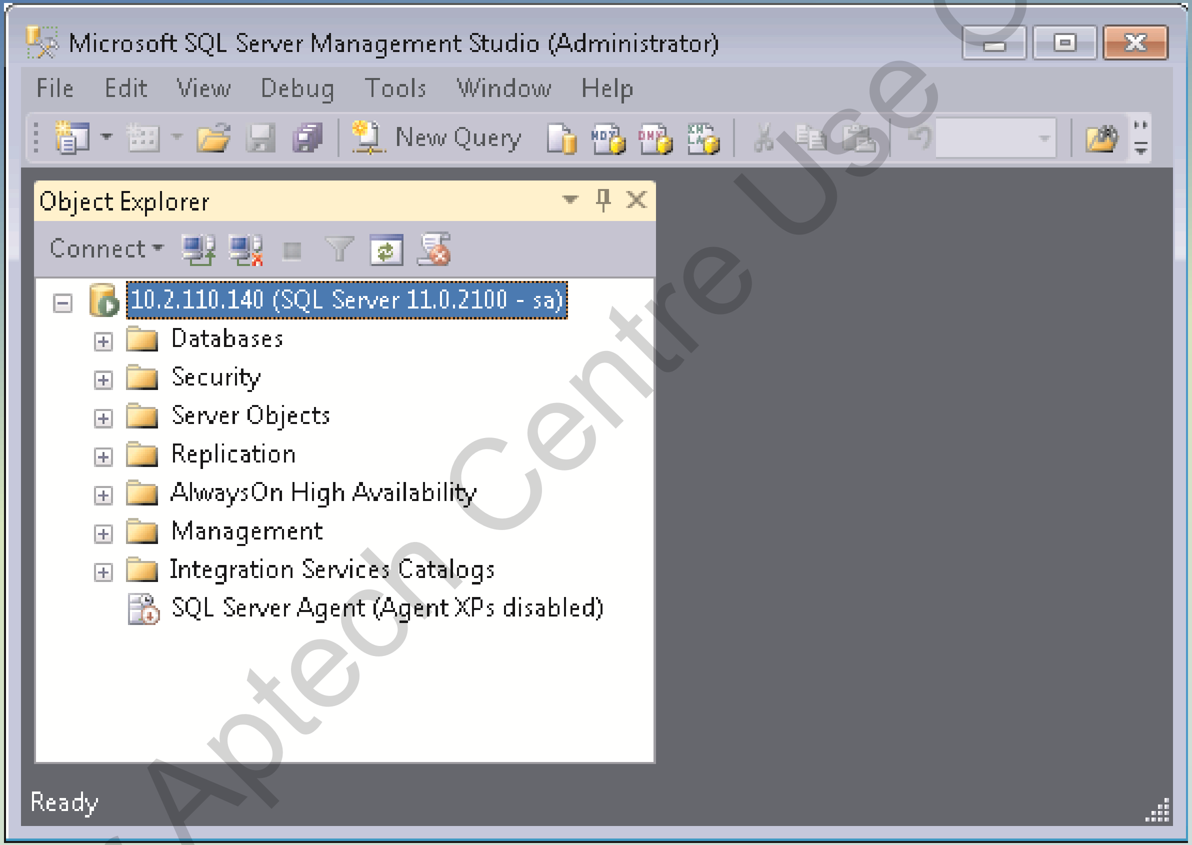
- Users are not allowed to directly update the information in system database objects, such as system tables, system stored procedures, and catalog views.
- However, users can avail a complete set of administrative tools allowing them to fully administer the system and manage all users and database objects.
- These are as follows:

Administration Utilities:

- From SQL Server 2005 onwards, several SQL Server administrative utilities are integrated into SSMS.
- It is the core administrative console for SQL Server installations.
- It enables to perform high-level administrative functions, schedule routine maintenance tasks, and so forth.

Modifying System Data 2-3

- Following figure shows the SQL Server 2012 Management Studio window:


Microsoft SQL Server Management Studio (Administrator)
File Edit View Debug Tools Window Help
New Query
Object Explorer
Connect
10.2.110.140 (SQL Server 11.0.2100 - sa)
Databases
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)
Ready

SQL Server Management Objects (SQL-SMO) API:

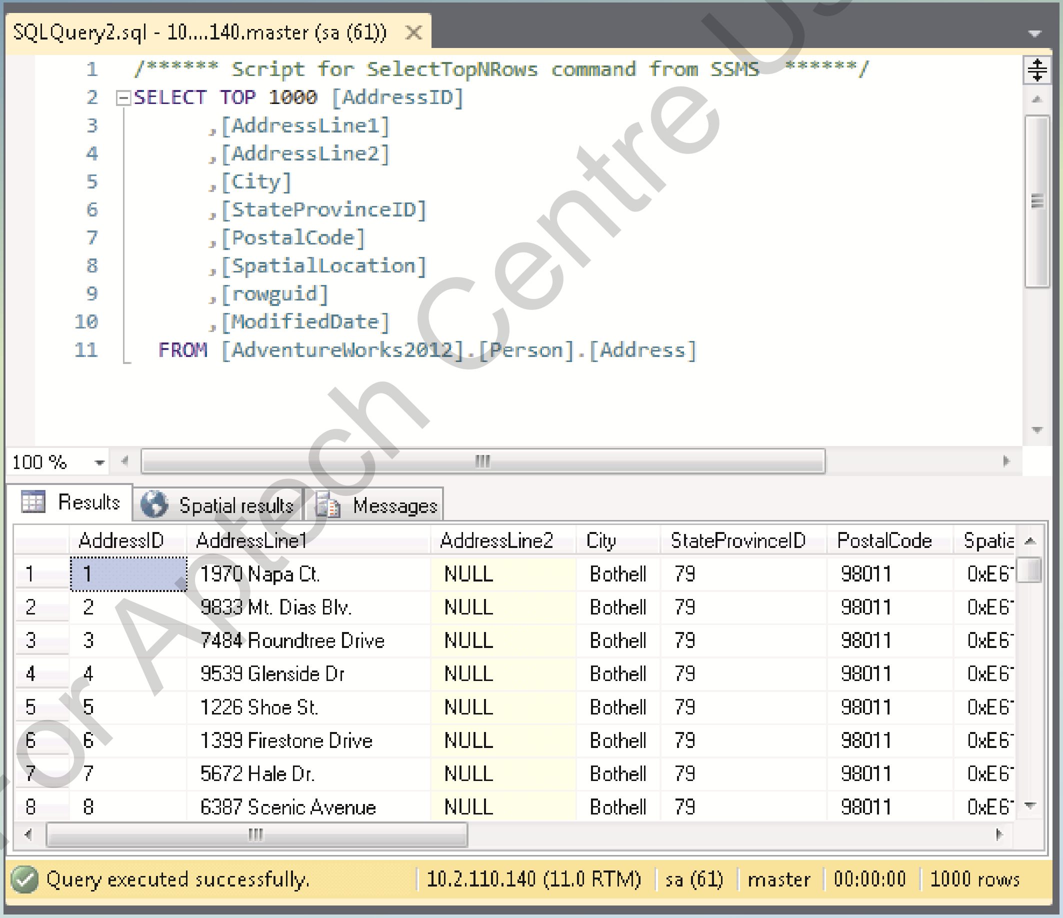
- Includes complete functionality for administering SQL Server in applications.

SQL Server 2012

Modifying System Data 3-3

Transact-SQL scripts and stored procedures:

- These use system stored procedures and Transact-SQL DDL statements. Following figure shows a Transact-SQL query window:



SQLQuery2.sql - 10....140.master (sa (61))

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP 1000 [AddressID]
      ,[AddressLine1]
      ,[AddressLine2]
      ,[City]
      ,[StateProvinceID]
      ,[PostalCode]
      ,[SpatialLocation]
      ,[rowguid]
      ,[ModifiedDate]
  FROM [AdventureWorks2012].[Person].[Address]
```

	AddressID	AddressLine1	AddressLine2	City	StateProvinceID	PostalCode	Spatia
1	1	1970 Napa Ct.	NULL	Bothell	79	98011	0xE6
2	2	9833 Mt. Dias Blv.	NULL	Bothell	79	98011	0xE6
3	3	7484 Roundtree Drive	NULL	Bothell	79	98011	0xE6
4	4	9539 Glenside Dr	NULL	Bothell	79	98011	0xE6
5	5	1226 Shoe St.	NULL	Bothell	79	98011	0xE6
6	6	1399 Firestone Drive	NULL	Bothell	79	98011	0xE6
7	7	5672 Hale Dr.	NULL	Bothell	79	98011	0xE6
8	8	6387 Scenic Avenue	NULL	Bothell	79	98011	0xE6

Query executed successfully. | 10.2.110.140 (11.0 RTM) | sa (61) | master | 00:00:00 | 1000 rows

SQL Server 2012

Viewing System Database Data

- Database applications can determine catalog and system information by using any of these approaches:

System catalog views

- Views displaying metadata for describing database objects in an SQL Server instance.

SQL-SMO

- New managed code object model, providing a set of objects used for managing Microsoft SQL Server.

Catalog functions, methods, attributes, or properties of the data API

- Used in ActiveX Data Objects (ADO), OLE DB, or ODBC applications.

Stored Procedures and Functions

- Used in Transact-SQL as stored procedures and built-in functions.

SQL Server 2012

Creating Databases 1-4

- To create a user-defined database, the information required is as follows:

Name of the database

Owner or creator of the database

Size of the database

Files and filegroups used to store it

Creating Databases 2-4

- The syntax to create a user-defined database is as follows:

Syntax:

```
CREATE DATABASE DATABASE_NAME
[ ON
[ PRIMARY ] [ <filespec> [ ,...n ]
[ , <filegroup> [ ,...n ] ]
[ LOG ON { <filespec> [ ,...n ] } ]
]
[ COLLATE collation_name ]
]
[;]
```

where,

`DATABASE_NAME`: is the name of the database to be created.

`ON`: indicates the disk files to be used to store the data sections of the database and data files.

`PRIMARY`: is the associated `<filespec>` list defining the primary file.

`<filespec>`: controls the file properties.

`<filegroup>`: controls filegroup properties.

SQL Server 2012

Creating Databases 3-4

`LOG ON`: indicates disk files to be used for storing the database log and log files.
`COLLATE collation_name`: is the default collation for the database. A collation defines rules for comparing and sorting character data based on the standard of particular language and locale. Collation name can be either a Windows collation name or a SQL collation name.

- Following code snippet shows how to create a database with database file and transaction log file with collation name:

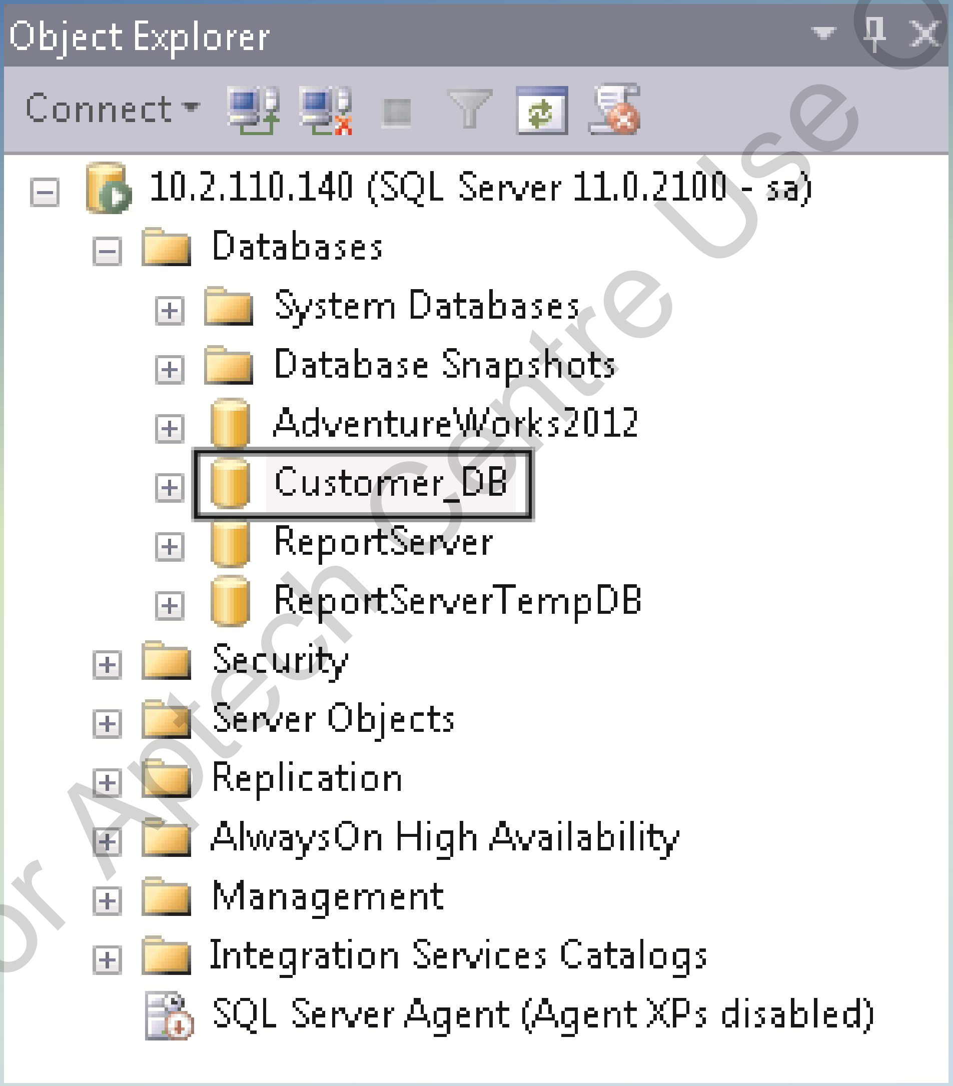
```
CREATE DATABASE [Customer_DB] ON PRIMARY
( NAME = 'Customer_DB', FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Customer_DB.mdf')
LOG ON
( NAME = 'Customer_DB_log', FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Customer_DB_log.ldf')
COLLATE SQL_Latin1_General_CP1_CI_AS
```

- After executing the code, SQL Server 2012 displays the message 'Command(s) completed successfully‘.

SQL Server 2012

Creating Databases 4-4

- Following figure shows the database **`Customer_DB`** listed in the **Object Explorer**:



Object Explorer

Connect

```
10.2.110.140 (SQL Server 11.0.2100 - sa)
  Databases
    System Databases
    Database Snapshots
    AdventureWorks2012
    Customer_DB
    ReportServer
    ReportServerTempDB
  Security
  Server Objects
  Replication
  AlwaysOn High Availability
  Management
  Integration Services Catalogs
  SQL Server Agent (Agent XPs disabled)
```

Modifying Databases 1-3

- As a user-defined database grows or diminishes, the database size will be expanded or be shrunk automatically or manually.
- The syntax to modify a database is as follows:

Syntax:

```
ALTER DATABASE database_name
{
<add_or_modify_files>
| <add_or_modify_filegroups>
| <set_database_options>
| MODIFY NAME = new_database_name
| COLLATE collation_name
}
[;]
```

where,

`database_name`: is the original name of the database.

`MODIFY NAME = new_database_name`: is the new name of the database to which it is to be renamed.

`COLLATE collation_name`: is the collation name of the database.

Modifying Databases 2-3

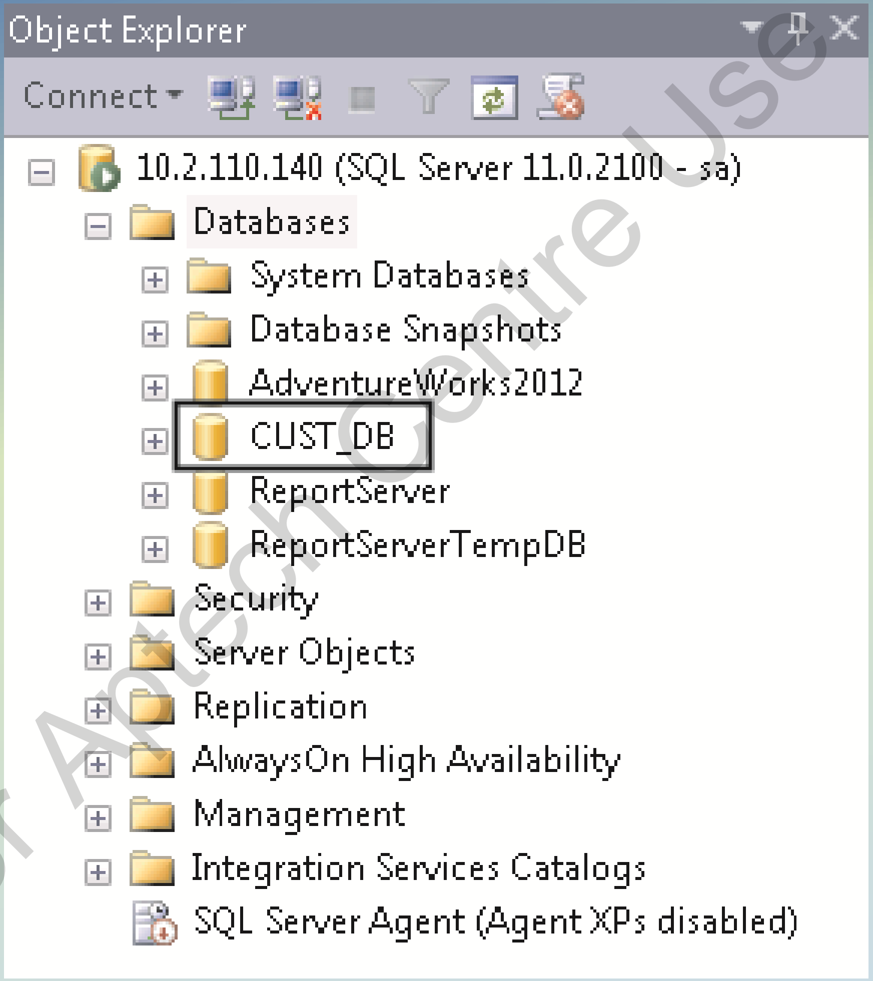
`<add_or_modify_files>`: is the file to be added, removed, or modified.
`<add_or_modify_filegroups>`: is the filegroup to be added, modified, or removed from the database.
`<set_database_options>`: is the database-level option influencing the characteristics of the database that can be set for each database. These options are unique to each database and do not affect other databases.

- Following code snippet shows how to rename a database **`Customer_DB`** with a new database name, **`CUST_DB`**:

```
ALTER DATABASE Customer_DB MODIFY NAME = CUST_DB
```

Modifying Databases 3-3

- Following figure shows database **`Customer_DB`** is renamed with a new database name, **`CUST_DB`**:



10.2.110.140 (SQL Server 11.0.2100 - sa)
Databases
System Databases
Database Snapshots
AdventureWorks2012
CUST_DB
ReportServer
ReportServerTempDB
Security
Server Objects
Replication
AlwaysOn High Availability
Management
Integration Services Catalogs
SQL Server Agent (Agent XPs disabled)

SQL Server 2012

Ownership of Databases 1-2

- In SQL Server 2012, the ownership of a user-defined database can be changed.
- Ownership of system databases cannot be changed.
- The system procedure `sp_changedbowner` is used to change the ownership of a database. The syntax is as follows:

Syntax:

```
sp_changedbowner [ @loginame = ] 'login'
```

where,

`login`: is an existing database username.

SQL Server 2012

Ownership of Databases 2-2

- After `sp_changedbowner` is executed, the new owner is known as the `dbo` user inside the selected database.
- The `dbo` receives permissions to perform all activities in the database.
- The owner of the `master`, `model`, or `tempdb` system databases cannot be changed.
- Following code snippet, when executed, makes the login '`sa`' the owner of the current database and maps '`sa`' to existing aliases that are assigned to the old database owner, and will display 'Command(s) completed successfully‘:

```
USE CUST_DB
EXEC sp_changedbowner 'sa’
```

Setting Database Options 1-2

- Database-level options determine the characteristics of the database that can be set for each database.
- These options are unique to each database, so they do not affect other databases.
- These database options are set to default values when a database is first created, and can then, be changed by using the `SET` clause of the `ALTER DATABASE` statement.
- Following table shows the database options that are supported by SQL Server 2012:

Option Type	Description
Automatic options	Controls automatic behavior of database.
Cursor options	Controls cursor behavior.
Recovery options	Controls recovery models of database.
Miscellaneous options	Controls ANSI compliance.
State options	Controls state of database, such as online/offline and user connectivity.

SQL Server 2012

Setting Database Options 2-2

- Following code snippet when executed sets `AUTO_SHRINK` option for the `CUST_DB` database to `ON`:

```
USE CUST_DB;
ALTER DATABASE CUST_DB
SET AUTO_SHRINK ON
```

- The `AUTO_SHRINK` options when set to `ON`, shrinks the database that have free space.

AdventureWorks2012 Database 1-2

- The `AdventureWorks2012` database consists of around 100 features.
- Some of the key features are as follows:

- Database Engine
- Analysis Services
- Integration Services
- Notification Services
- Reporting Services
- Replication Facilities
- A set of integrated samples for two multiple feature-based samples: `HRResume` and `Storefront`.

AdventureWorks2012 Database 2-2

- The sample database consists of these parts:

- **AdventureWorks2012: Sample OLTP database**
- **AdventureWorks2012DW: Sample Data warehouse**
- **AdventureWorks2012AS: Sample Analysis Services database**

SQL Server 2012

Filegroups 1-2

In SQL Server, data files are used to store database files. The data files are further subdivided into filegroups for the sake of performance.

Each filegroup is used to group related files that together store a database object.

Every database has a primary filegroup by default. This filegroup contains the primary data file.

The primary file group and data files are created automatically with default property values at the time of creation of the database.

User-defined filegroups can then be created to group data files together for administrative, data allocation, and placement purposes.

SQL Server 2012

Filegroups 2-2

For example, three files named `Customer_Data1.ndf`, `Customer_Data2.ndf`, and `Customer_Data3.ndf` can be created on three disk drives respectively.

These can then be assigned to the filegroup `Customer_fgroup1`. A table can then be created specifically on the filegroup `Customer_fgroup1`.

Queries for data from the table will be spread across the three disk drives thereby, improving performance.

- Following table shows the filegroups that are supported by SQL Server 2012:

Filegroup	Description
Primary	The filegroup that consists of the primary file. All system tables are placed inside the primary filegroup.
User-defined	Any filegroup that is created by the user at the time of creating or modifying databases.

SQL Server 2012

Adding Filegroups to an Existing Database 1-5

- Filegroups can be created when the database is created for the first time or can be created later when more files are added to the database.
- However, files cannot be moved to a different filegroup after the files have been added to the database.
- A file cannot be a member of more than one filegroup at the same time.
- A maximum of 32,767 filegroups can be created for each database.
- Filegroups can contain only data files. Transaction log files cannot belong to a filegroup.

SQL Server 2012

Adding Filegroups to an Existing Database 2-5

- The following is the syntax to add filegroups while creating a database:

Syntax:

```
CREATE DATABASE database_name
[ ON
[ PRIMARY ] [ <filespec> [ ,...n ]
[ , <filegroup> [ ,...n ] ]
[ LOG ON { <filespec> [ ,...n ] } ]
]
[ COLLATE collation_name ]
]
[;]
```

where,

`database_name`: is the name of the new database.

`ON`: indicates the disk files to store the data sections of the database, and data files.

`PRIMARY and associated <filespec> list`: define the primary file. The first file specified in the <filespec> entry in the primary filegroup becomes the primary file.

`LOG ON`: indicates the disk files used to store the database log files.

`COLLATE collation_name`: is the default collation for the database.

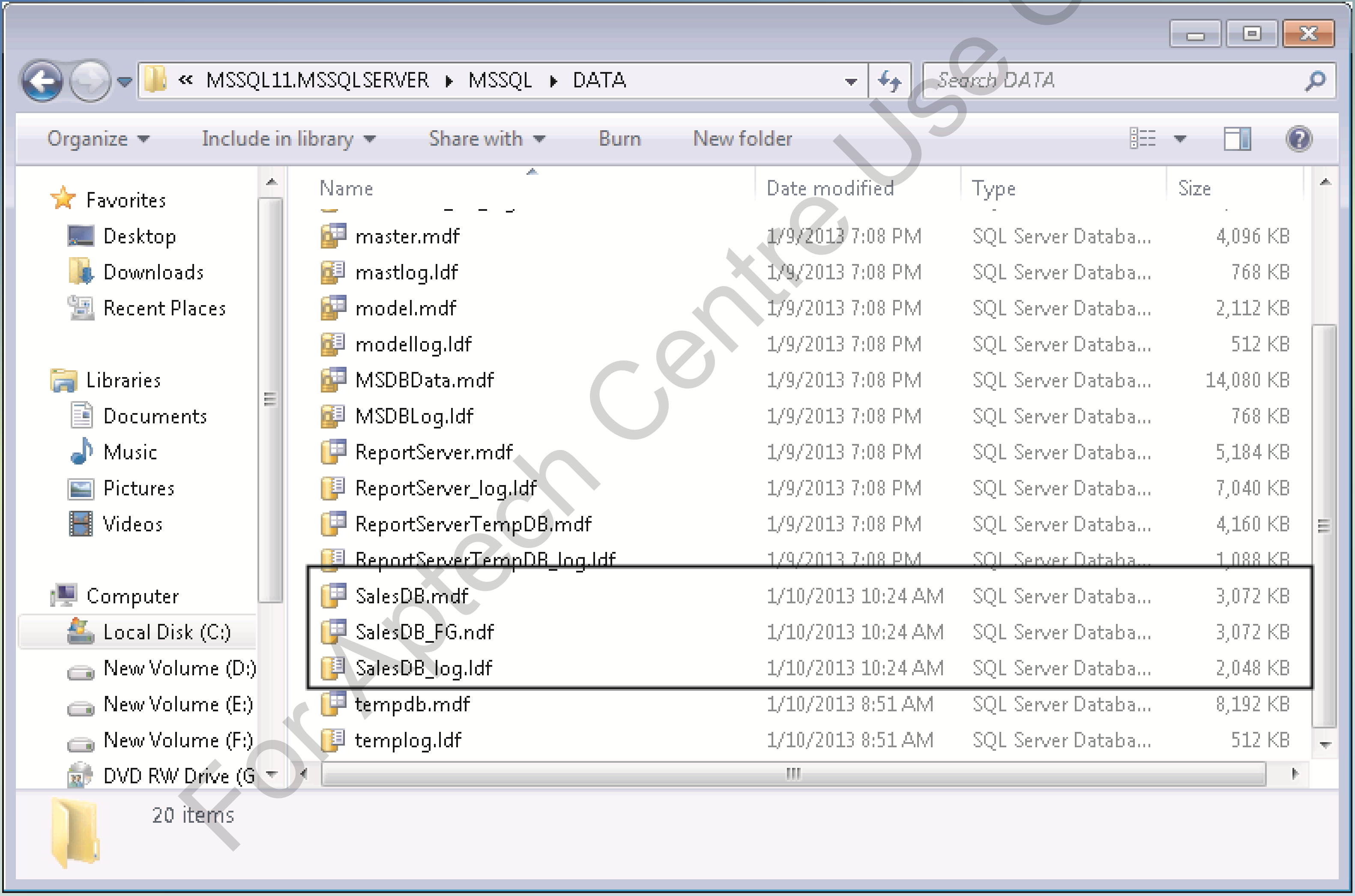
Adding Filegroups to an Existing Database 3-5

- Following code snippet shows how to add a filegroup (`PRIMARY` as default) while creating a database, called **`SalesDB`**:

```
CREATE DATABASE [SalesDB] ON PRIMARY
( NAME = 'SalesDB', FILENAME ='C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\SalesDB.mdf' , SIZE = 3072KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB ),
FILEGROUP [MyFileGroup]
( NAME = 'SalesDB_FG', FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\SalesDB_FG.ndf' , SIZE = 3072KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = 'SalesDB_log', FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\SalesDB_log.ldf' , SIZE = 2048KB ,
MAXSIZE = 2048GB , FILEGROWTH = 10%)
COLLATE SQL_Latin1_General_CP1_CI_AS
```

Adding Filegroups to an Existing Database 4-5

- Following figure shows the file groups when creating **SalesDB** database:



Name	Date modified	Type	Size
master.mdf	1/9/2013 7:08 PM	SQL Server Databa...	4,096 KB
mastlog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	768 KB
model.mdf	1/9/2013 7:08 PM	SQL Server Databa...	2,112 KB
modellog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	512 KB
MSDBData.mdf	1/9/2013 7:08 PM	SQL Server Databa...	14,080 KB
MSDBLog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	768 KB
ReportServer.mdf	1/9/2013 7:08 PM	SQL Server Databa...	5,184 KB
ReportServer_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	7,040 KB
ReportServerTempDB.mdf	1/9/2013 7:08 PM	SQL Server Databa...	4,160 KB
ReportServerTempDB_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	1,088 KB
SalesDB.mdf	1/10/2013 10:24 AM	SQL Server Databa...	3,072 KB
SalesDB_FG.ndf	1/10/2013 10:24 AM	SQL Server Databa...	3,072 KB
SalesDB_log.ldf	1/10/2013 10:24 AM	SQL Server Databa...	2,048 KB
tempdb.mdf	1/10/2013 8:51 AM	SQL Server Databa...	8,192 KB
templog.ldf	1/10/2013 8:51 AM	SQL Server Databa...	512 KB

SQL Server 2012

Adding Filegroups to an Existing Database 5-5

- The syntax to add a filegroup to an existing database is as follows:

```
ALTER DATABASE database_name
{ <add_or_modify_files>
| <add_or_modify_filegroups>
| <set_database_options>
| MODIFY NAME = new_database_name
| COLLATE collation_name
}
[;]
```

- Following code snippet shows how to add a filegroup to an existing database, called **CUST_DB**:

```
USE CUST_DB;
ALTER DATABASE CUST_DB
ADD FILEGROUP FG_ReadOnly0
```

- After executing the code, SQL Server 2012 displays the message 'Command(s) completed successfully' and the filegroup `FG_ReadOnly` is added to the existing database, **CUST_DB**.

Default Filegroup 1-2

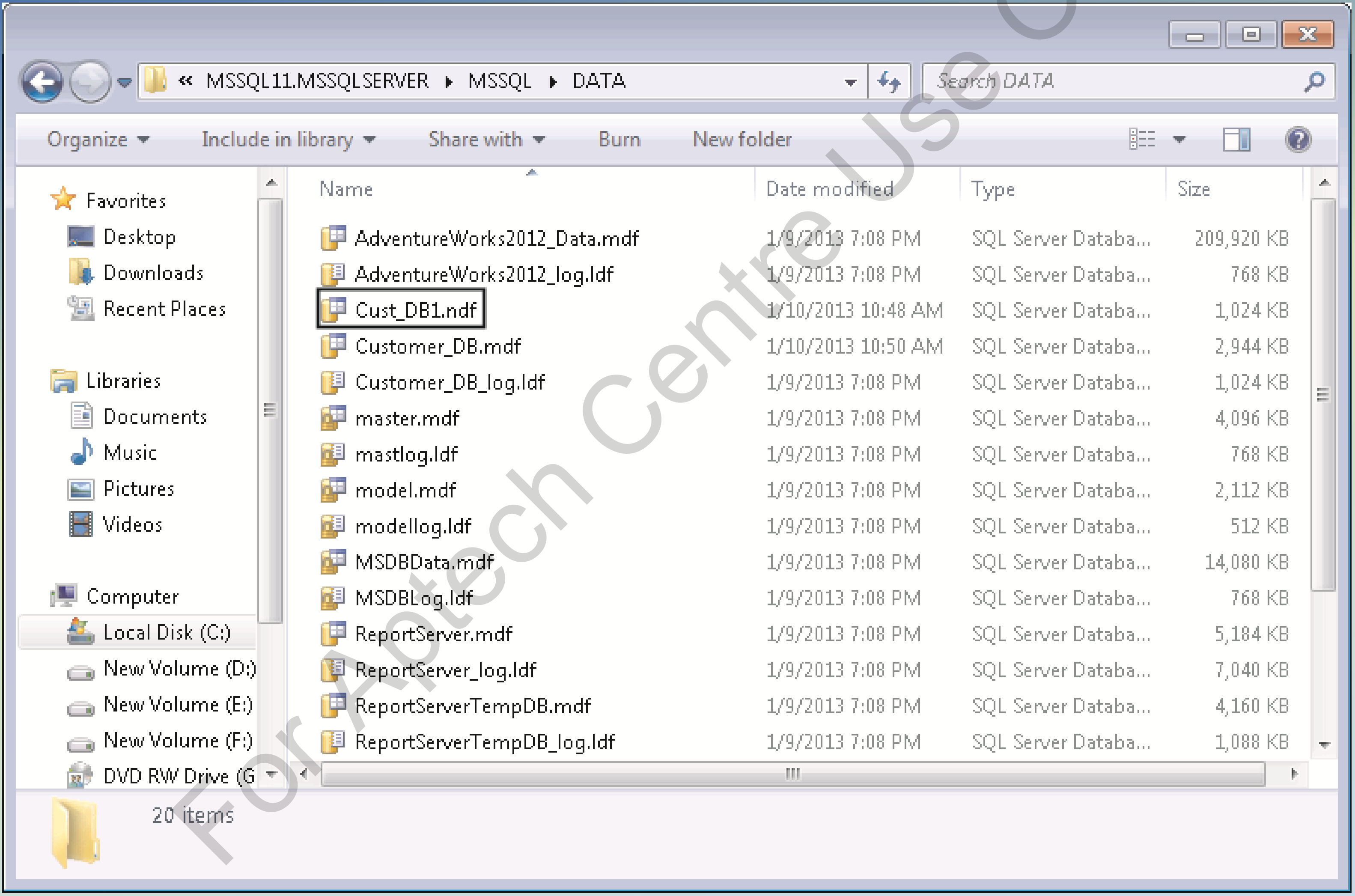
- Objects are assigned to the default filegroup when they are created in the database.
- The PRIMARY filegroup is the default filegroup. The default filegroup can be changed using the ALTER DATABASE statement.
- System objects and tables remain within the PRIMARY filegroup, but do not go into the new default filegroup.
- To make the FG_ReadOnly filegroup as default, it should contain at least one file inside it.
- Following code snippet shows how to create a new file, add it to the FG_ReadOnly filegroup, and make the FG_ReadOnly filegroup as the default filegroup:

```
USE CUST_DB
ALTER DATABASE CUST_DB
ADD FILE (NAME = Cust_DB1, FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Cust_DB1.ndf')
TO FILEGROUP FG_ReadOnly
ALTER DATABASE CUST_DB
MODIFY FILEGROUP FG_ReadOnly DEFAULT
```

- After executing the code, SQL Server 2012 displays the message saying the filegroup property 'DEFAULT' has been set.

Default Filegroup 2-2

- Following figure shows a new file **Cust_DB1** created:



MSSQL11.MSSQLSERVER ▸ MSSQL ▸ DATA

Name	Date modified	Type	Size
AdventureWorks2012_Data.mdf	1/9/2013 7:08 PM	SQL Server Databa...	209,920 KB
AdventureWorks2012_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	768 KB
Cust_DB1.ndf	1/10/2013 10:48 AM	SQL Server Databa...	1,024 KB
Customer_DB.mdf	1/10/2013 10:50 AM	SQL Server Databa...	2,944 KB
Customer_DB_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	1,024 KB
master.mdf	1/9/2013 7:08 PM	SQL Server Databa...	4,096 KB
mastlog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	768 KB
model.mdf	1/9/2013 7:08 PM	SQL Server Databa...	2,112 KB
modellog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	512 KB
MSDBData.mdf	1/9/2013 7:08 PM	SQL Server Databa...	14,080 KB
MSDBLog.ldf	1/9/2013 7:08 PM	SQL Server Databa...	768 KB
ReportServer.mdf	1/9/2013 7:08 PM	SQL Server Databa...	5,184 KB
ReportServer_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	7,040 KB
ReportServerTempDB.mdf	1/9/2013 7:08 PM	SQL Server Databa...	4,160 KB
ReportServerTempDB_log.ldf	1/9/2013 7:08 PM	SQL Server Databa...	1,088 KB

20 items

Transaction Log 1-4

- A transaction log in SQL Server records all transactions and the database modifications made by each transaction.
- The transaction log is one of the critical components of the database.
- It can be the only source of recent data in case of system failure.
- The transaction logs support operations such as the following:

Recovery of individual transactions

- An incomplete transaction is rolled back in case of an application issuing a ROLLBACK statement or the Database Engine detecting an error.
- The log records are used to roll back the modifications.

Recovery of all incomplete transactions when SQL Server is started

- If a server that is running SQL Server fails, the databases may be left in an inconsistent state.
- When an instance of SQL Server is started, it runs a recovery of each database.

SQL Server 2012

Transaction Log 2-4

Rolling a restored database, file, filegroup, or page forward to the point of failure

- The database can be restored to the point of failure after a hardware loss or disk failure affecting the database files.

Supporting transactional replication

- The Log Reader Agent monitors the transaction log of each database configured for replications of transactions.

Supporting standby server solutions

- The standby-server solutions, database mirroring, and log shipping depend on the transaction log.

Transaction Log 3-4

Working of Transaction Logs:

- A database in SQL Server 2012 has at least one data file and one transaction log file.
- Data and transaction log information are kept separated on the same file.
- Individual files are used by only one database.
- SQL Server uses the transaction log of each database to recover transactions.
- The transaction log is a serial record of all modifications that have occurred in the database as well as the transactions that performed the modifications.
- This log keeps enough information to undo the modifications made during each transaction.
- The transaction log records the allocation and deallocation of pages and the commit or rollback of each transaction.
- This feature enables SQL Server either to roll forward or to back out.

Transaction Log 4-4

- The rollback of each transaction is executed using the following ways:

A transaction is rolled forward when a transaction log is applied.

A transaction is rolled back when an incomplete transaction is backed out.

Adding Log files to a database:

- The syntax to modify a database and add log files is as follows:

Syntax:

```
ALTER DATABASE database_name
{
...
}
[;]
```

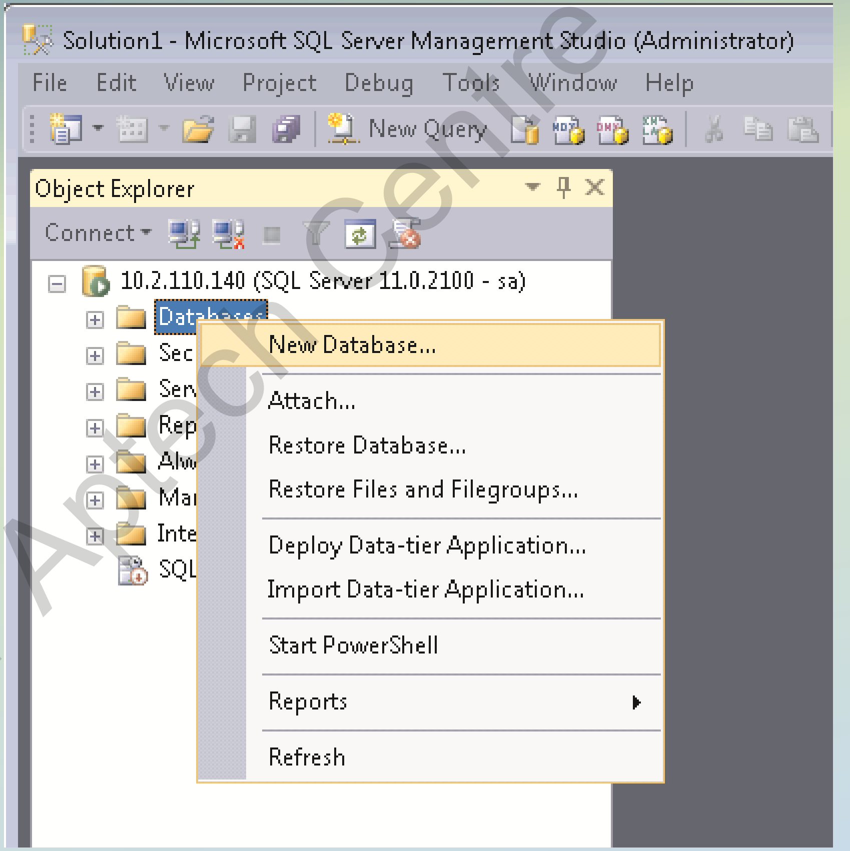
```
<add_or_modify_files>::=
{
ADD FILE <filespec> [ ,...n ]
[ TO FILEGROUP { filegroup_name | DEFAULT } ]
| ADD LOG FILE <filespec> [ ,...n ]
| REMOVE FILE logical_file_name
| MODIFY FILE <filespec>
}
```

Create Database Procedure 1-7

- The steps to create a database using SSMS are as follows:

1
- In **Object Explorer**, connect to an instance of the SQL Server Database Engine and then, expand that instance.

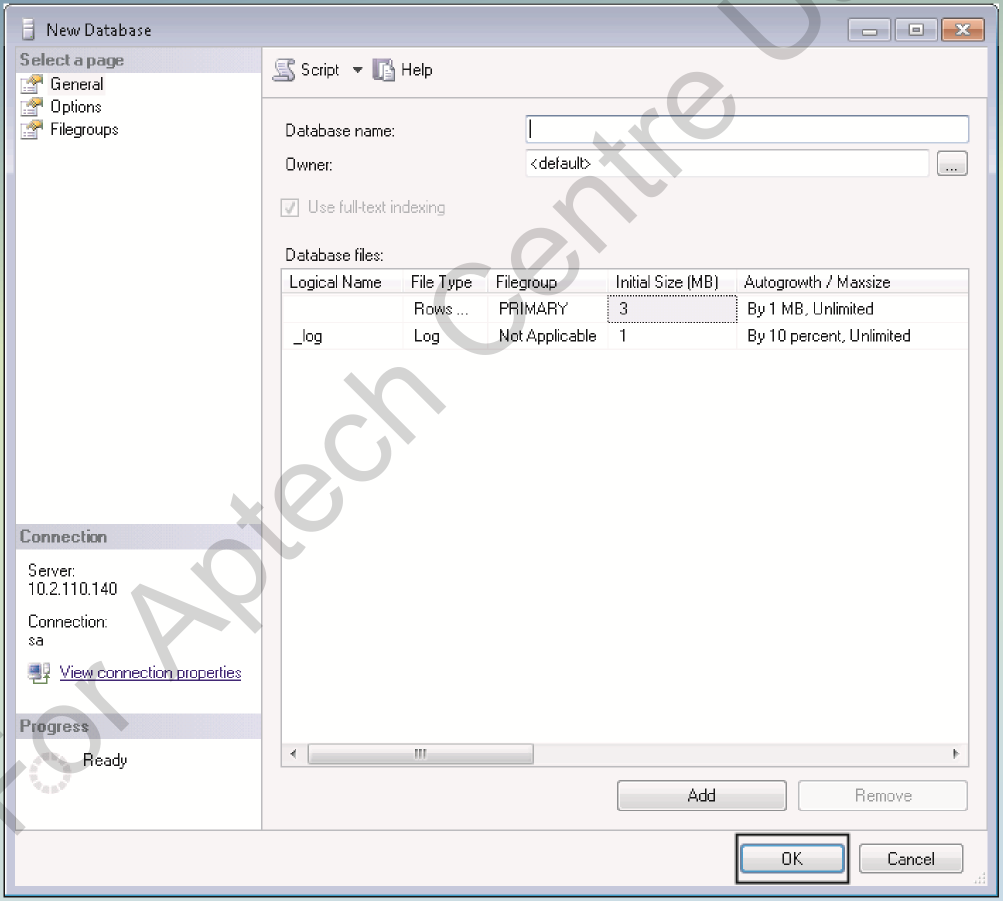
2
- Right-click **Databases**, and then, click **New Database** as shown in the following figure:


Solution1 - Microsoft SQL Server Management Studio (Administrator)
File Edit View Project Debug Tools Window Help
New Query
Object Explorer
Connect
10.2.110.140 (SQL Server 11.0.2100 - sa)
Databases
Sec
Serv
Rep
Alw
Mai
Inte
SQL
New Database...
Attach...
Restore Database...
Restore Files and Filegroups...
Deploy Data-tier Application...
Import Data-tier Application...
Start PowerShell
Reports
Refresh

Create Database Procedure 2-7

3
- In **New Database**, enter a database name.

4
- To create the database by accepting all default values, click **OK**, as shown in the following figure; otherwise, continue with the following optional steps:



Database name:

Owner: <default>

☑ Use full-text indexing

Database files:

Logical Name	File Type	Filegroup	Initial Size (MB)	Autogrowth / Maxsize
	Rows ...	PRIMARY	3	By 1 MB, Unlimited
_log	Log	Not Applicable	1	By 10 percent, Unlimited

Server:
10.2.110.140

Connection:
sa

View connection properties

Ready

SQL Server 2012

Create Database Procedure 3-7

5

- To change the owner name, click (…) to select another owner.

6

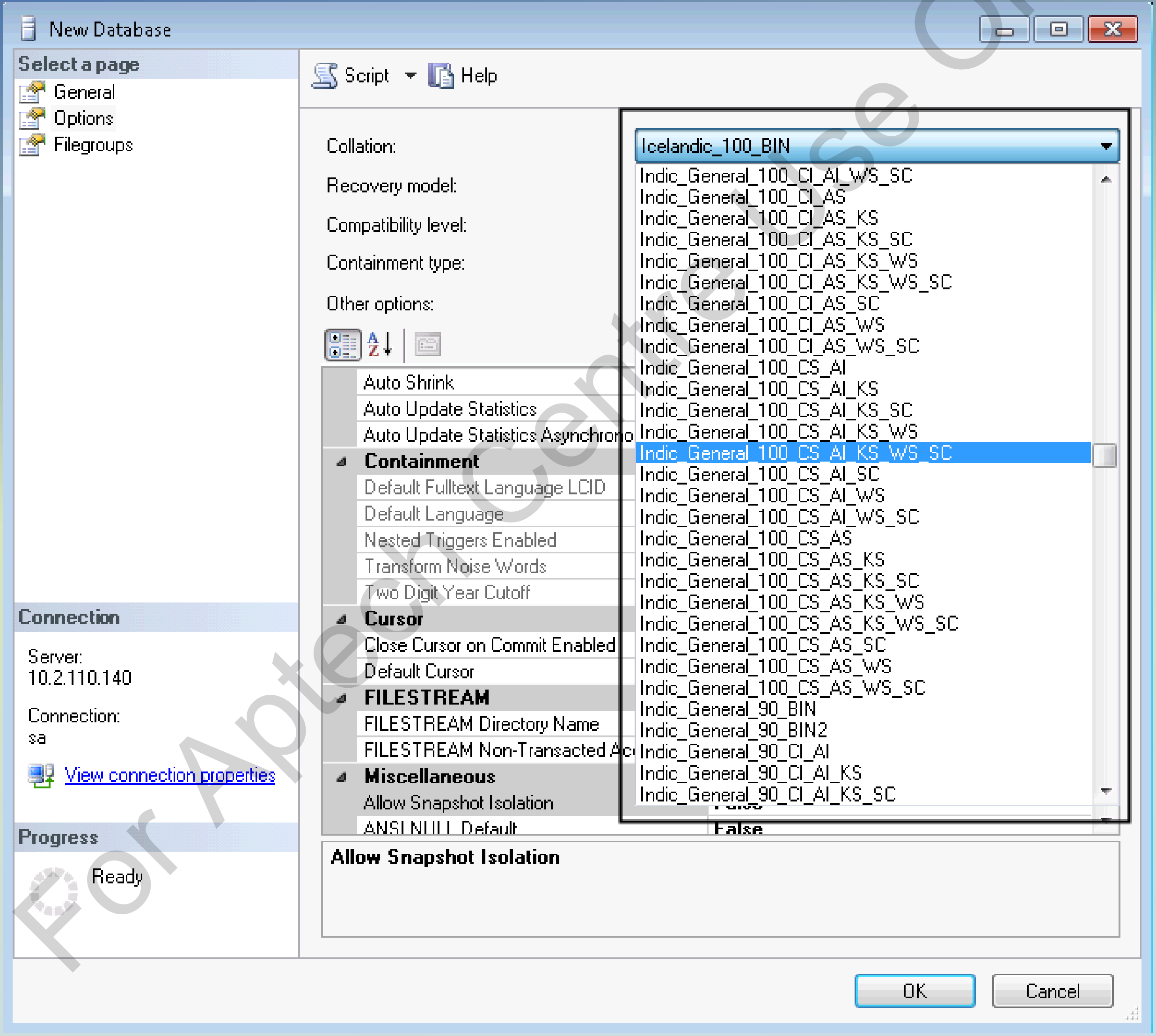
- To change the default values of the primary data and transaction log files, in the Database files grid, click the appropriate cell and enter the new value.

7

- To change the collation of the database, select the **Options** page, and then, select a collation from the list.

Create Database Procedure 4-7

- This is shown in the following figure:



New Database

Select a page
- General
- Options
- Filegroups

Script ▾ Help

Collation: Icelandic_100_BIN

Recovery model:

Compatibility level:

Containment type:

Other options:

Indic_General_100_CI_AI_WS_SC
Indic_General_100_CI_AS
Indic_General_100_CI_AS_KS
Indic_General_100_CI_AS_KS_SC
Indic_General_100_CI_AS_KS_WS
Indic_General_100_CI_AS_KS_WS_SC
Indic_General_100_CI_AS_SC
Indic_General_100_CI_AS_WS
Indic_General_100_CI_AS_WS_SC
Indic_General_100_CS_AI
Indic_General_100_CS_AI_KS
Indic_General_100_CS_AI_KS_SC
Indic_General_100_CS_AI_KS_WS
Indic_General_100_CS_AI_KS_WS_SC
Indic_General_100_CS_AI_SC
Indic_General_100_CS_AI_WS
Indic_General_100_CS_AI_WS_SC
Indic_General_100_CS_AS
Indic_General_100_CS_AS_KS
Indic_General_100_CS_AS_KS_SC
Indic_General_100_CS_AS_KS_WS
Indic_General_100_CS_AS_KS_WS_SC
Indic_General_100_CS_AS_SC
Indic_General_100_CS_AS_WS
Indic_General_100_CS_AS_WS_SC
Indic_General_90_BIN
Indic_General_90_BIN2
Indic_General_90_CI_AI
Indic_General_90_CI_AI_KS
Indic_General_90_CI_AI_KS_SC

Auto Shrink	
Auto Update Statistics	
Auto Update Statistics Asynchrono	
Containment	
Default Fulltext Language LCID	
Default Language	
Nested Triggers Enabled	
Transform Noise Words	
Two Digit Year Cutoff	
Cursor	
Close Cursor on Commit Enabled	
Default Cursor	
FILESTREAM	
FILESTREAM Directory Name	
FILESTREAM Non-Transacted Acc	
Miscellaneous	
Allow Snapshot Isolation	
ANSI NULL Default	False

Allow Snapshot Isolation

Connection

Server:
10.2.110.140

Connection:
sa

View connection properties

Progress

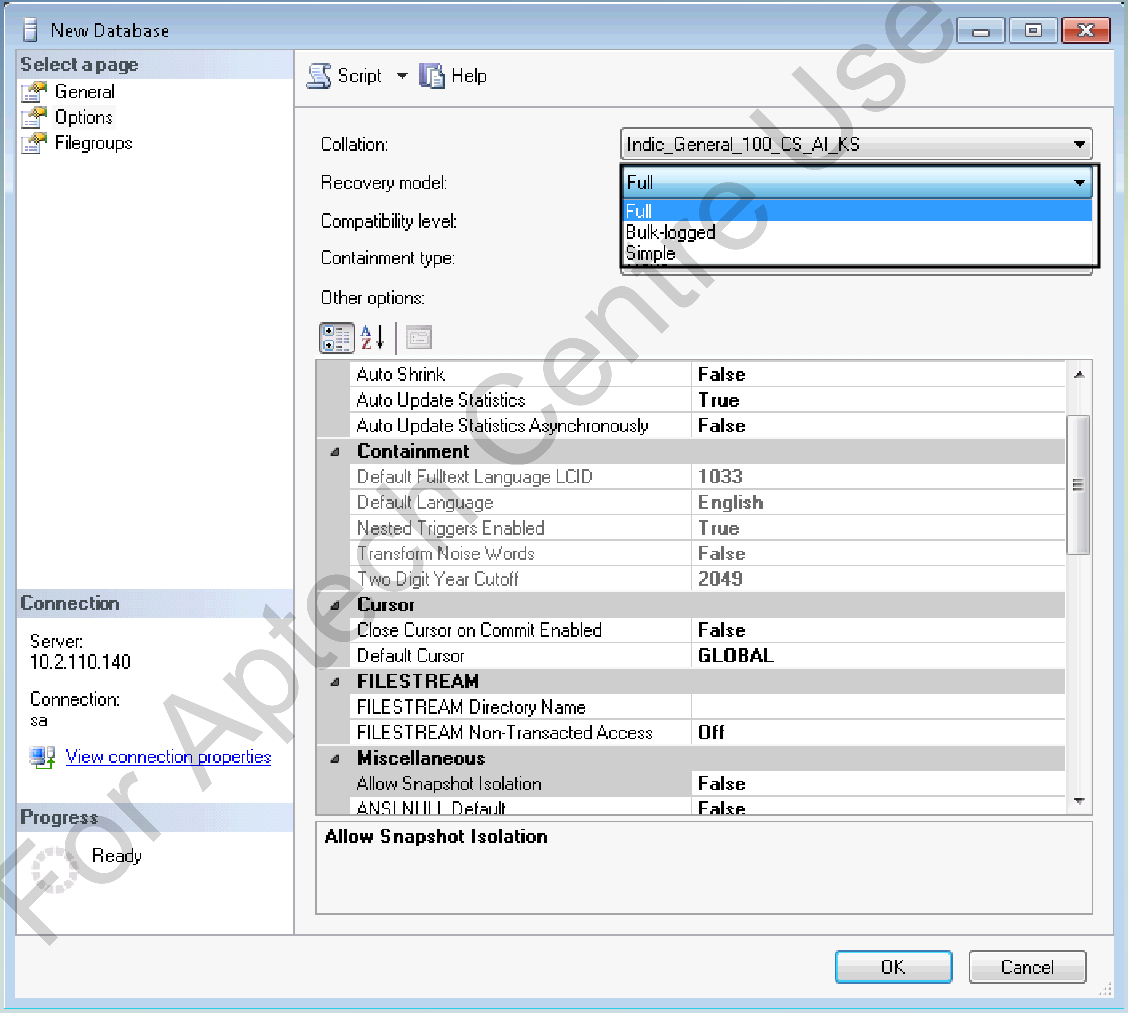
Ready

OK Cancel

Create Database Procedure 5-7

8

- To change the recovery model, select the **Options** page and then, select a recovery model from the list as shown in the following figure:



New Database

Select a page
- General
- Options
- Filegroups

Script | Help

Collation: Indic_General_100_CS_AI_KS

Recovery model: Full

- Full
- Bulk-logged
- Simple

Compatibility level:

Containment type:

Other options:

Auto Shrink	False
Auto Update Statistics	True
Auto Update Statistics Asynchronously	False
Containment	
Default Fulltext Language LCID	1033
Default Language	English
Nested Triggers Enabled	True
Transform Noise Words	False
Two Digit Year Cutoff	2049
Cursor	
Close Cursor on Commit Enabled	False
Default Cursor	GLOBAL
FILESTREAM	
FILESTREAM Directory Name	
FILESTREAM Non-Transacted Access	Off
Miscellaneous	
Allow Snapshot Isolation	False
ANSI NULL Default	False

Allow Snapshot Isolation

Connection

Server:
10.2.110.140

Connection:
sa

View connection properties

Progress

Ready

OK | Cancel

SQL Server 2012

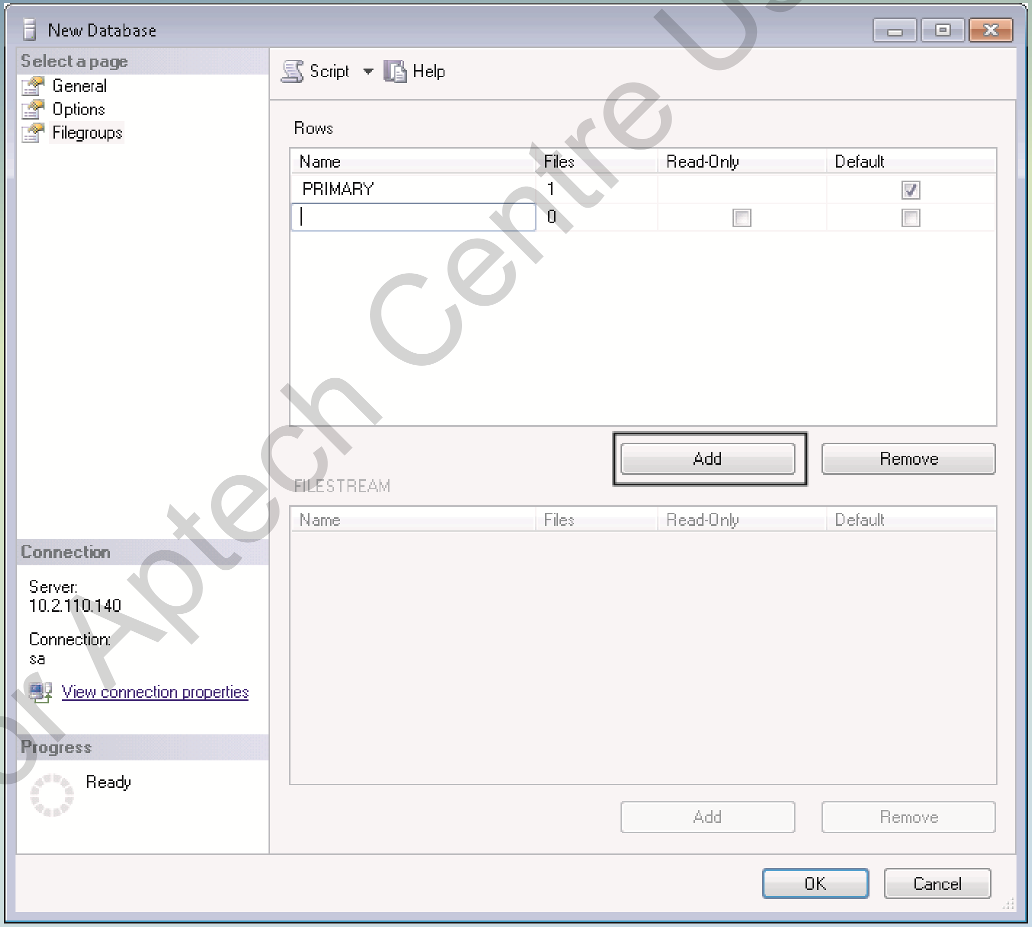
Create Database Procedure 6-7

9

- To change database options, select the **Options** page, and then, modify the database options.

10

- To add a new filegroup, click the **Filegroups** page. Click **Add** and then, enter the values for the filegroup as shown in the following figure:



Server:
10.2.110.140

Connection:
sa

Rows

Name	Files	Read-Only	Default
PRIMARY	1		✓
	0	☐	☐

FILESTREAM

Name	Files	Read-Only	Default

SQL Server 2012

Create Database Procedure 7-7

11

- To add an extended property to the database, select the **Extended Properties** page.
- In the **Name** column, enter a name for the extended property.
- In the **Value** column, enter the extended property text. For example, enter one or more statements that describe the database.

12

- To create the database, click **OK**.

Types of Database Modification Methods

- Following table lists and describes types of modifications of databases and modification methods:

Type of Modifications	Modification Methods
Increasing the size of a database	ALTER DATABASE statement or the database properties in SSMS
Changing the physical location of a database	ALTER DATABASE statement
Adding data or transaction log files	ALTER DATABASE statement or the database properties in SSMS
Shrinking a database	DBCC SHRINKDATABASE statement or the Shrink Database option in SSMS, accessed through the node for the specific database
Shrinking a database file	DBCC SHRINKFILE statement
Deleting data or log files	ALTER DATABASE statement or the database properties in SSMS
Adding a filegroup to a database	ALTER DATABASE statement or the database properties in SSMS
Changing the default filegroup	ALTER DATABASE statement
Changing database options	ALTER DATABASE statement or the database properties in SSMS
Changing the database owner	sp_changedbowner system stored procedure

SQL Server 2012

Drop Database Procedure 1-4

- A full backup of the database needs to be taken before dropping a database.
- A deleted database can be re-created only by restoring a backup.
- The steps to delete or drop a database using SSMS are as follows:

1

- In **Object Explorer**, connect to an instance of the SQL Server Database Engine, and then, expand that instance.

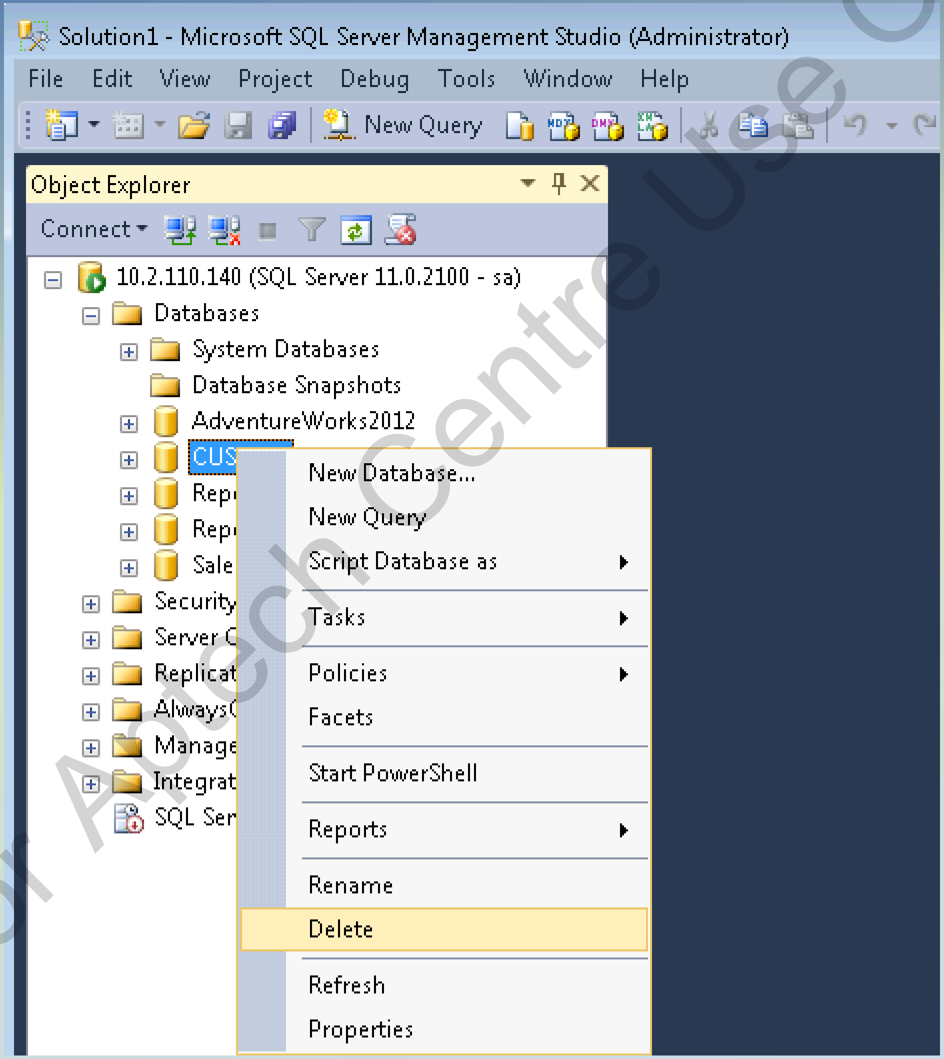
2

- Expand **Databases**, right-click the database to delete, and then, click **Delete**.

SQL Server 2012

Drop Database Procedure 2-4

➢ This is shown in the following figure:


Solution1 - Microsoft SQL Server Management Studio (Administrator)

Drop Database Procedure 3-4

3

- Confirm that the correct database is selected, and then, click **OK**.

➢ The syntax to delete or drop a database using Transact-SQL is as follows:

```
CREATE DATABASE database_snapshot_name
ON
(
NAME = logical_file_name,
FILENAME = 'os_file_name'
) [ ,...n ]
AS SNAPSHOT OF source_database_name
[;]
```

where,

`database_snapshot_name`: is the name of the new database snapshot.

`ON ( NAME = logical_file_name, FILENAME = 'os_file_name' ) [ ,... n ]`: is the list of files in the source database. For the snapshot to work, all the data files must be specified individually.

Drop Database Procedure 4-4

`AS SNAPSHOT OF source_database_name`: is the database being created is a database snapshot of the source database specified by `source_database_name`.

- Following code snippet creates a database snapshot on the **`CUST_DB`** database:

```
CREATE DATABASE customer_snapshot01 ON
( NAME = Customer_DB, FILENAME = 'C:\Program Files\Microsoft SQL
 Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\Customerdat_0100.ss')
AS SNAPSHOT OF CUST_DB;
```

SQL Server 2012

Summary

- An SQL Server database is made up of a collection of tables that stores sets of specific structured data.
- SQL Server 2012 supports three kinds of databases:
  - System databases
  - User-defined databases
  - Sample databases
- SQL Server uses system databases to support different parts of the DBMS.
- A fictitious company, Adventure Works Cycles is created as a scenario and the AdventureWorks2012 database is designed for this company.
- The SQL Server data files are used to store database files, which are further subdivided into filegroups for the sake of performance.
- Objects are assigned to the default filegroup when they are created in the database. The PRIMARY filegroup is the default filegroup.
- A database snapshot is a read-only, static view of a SQL Server database.