Final Exam

Stewart Johnston
CIS 150 – Intro to Database Administration
NCMC
johnstons1@student.ncmich.edu

December 11, 2018

Contents

List of Figures

1 Short Answer

1. Database schemas are spoken of in two senses. In an abstract formal sense, a schema is the definition of the relationships and data structures in a database. In the practical sense, a schema is often treated by DBMSs and DBAs as a namespace or container for database objects, such as tables, indices, etc. Users can be assigned permissions to a schema, and by default unless otherwise specified at user creation, they use the "dbo" schema.

2. Views are effectively virtual tables built from stored SELECT statements. They are useful for security primarily because permissions can be assigned for them such that users can't access the underlying data, and because they can represent arbitrary combinations of fields to fit the DBA's – or organization's – needs, omitting any information which is deemed sensitive or irrelevant to the view.

3. Database recovery methods include:

    - Database dumping, using tools or commands built into the DBMS to create a SQL script. Running this script will recreate both the schema definitions and the data in the DBMS.
    - Filesystem level backups, which require bringing the database offline while archives can be made of the files the DBMS uses to store the database. These can then be copied and brought back online.
    - Transaction or log-based backups. These use a combination of a file which can be copied, and the logs of actions performed on the database. The file can be brought back online, and an arbitrary selection of logs can be replayed on top of that file in chronological order. This is useful if there was some destructive action performed, and the database needed to be rewound to just before this happened.

4. If and only if fault tolerance is not a requirement, RAID 0 – which entails striping data across disks – makes for very fast parallel reading and writing of data. It requires only two disks to implement, the minimum size for any RAID set, but it is a fragile structure. Failure in either disk would make all data unrecoverable.

2 Longform questions

2.1 Security Weaknesses

SQL Server has many moving parts, several of which are not secure. These include:

- System Administrator and Service users in the database are by default granted to/based on accounts with similar roles in the OS itself. If, under any circumstance, the box on which the server lives is compromised the default behavior also leaves the server open to compromise. It is best practice to decouple as much of the server as possible from builtin OS accounts. The builtin admin OS account should ideally be disabled, if possible, and the users granted SysAdmin privileges shouldn't also have admin privileges with the same account for the RDBMS. In the same breath, the default access controls for new users and logins is quite permissive, and those defaults should be changed as soon as possible.

- Many components which serve a real use for SQL Server are also tainted with a history of being vulnerable. Services which live on the IIS webserver platform are chief among these. The webapps which use the database's information, reporting services, and other internet facing components should all be installed and run from separate machines, virtual or otherwise.

- SQL injection is usually an issue handled outside of the server's components, in the back end which receives user input. If all goes well, then all input should be sanitized and validated. Programmers being people, they are forgetful and make mistakes. This the case, all webapp facing procedures should use SQL mechanisms to quote input to remove its danger. QUOTENAME() is one such function, although its return type is limited.

2.2 Backup and Compress AdventureWorks

See figure 1.

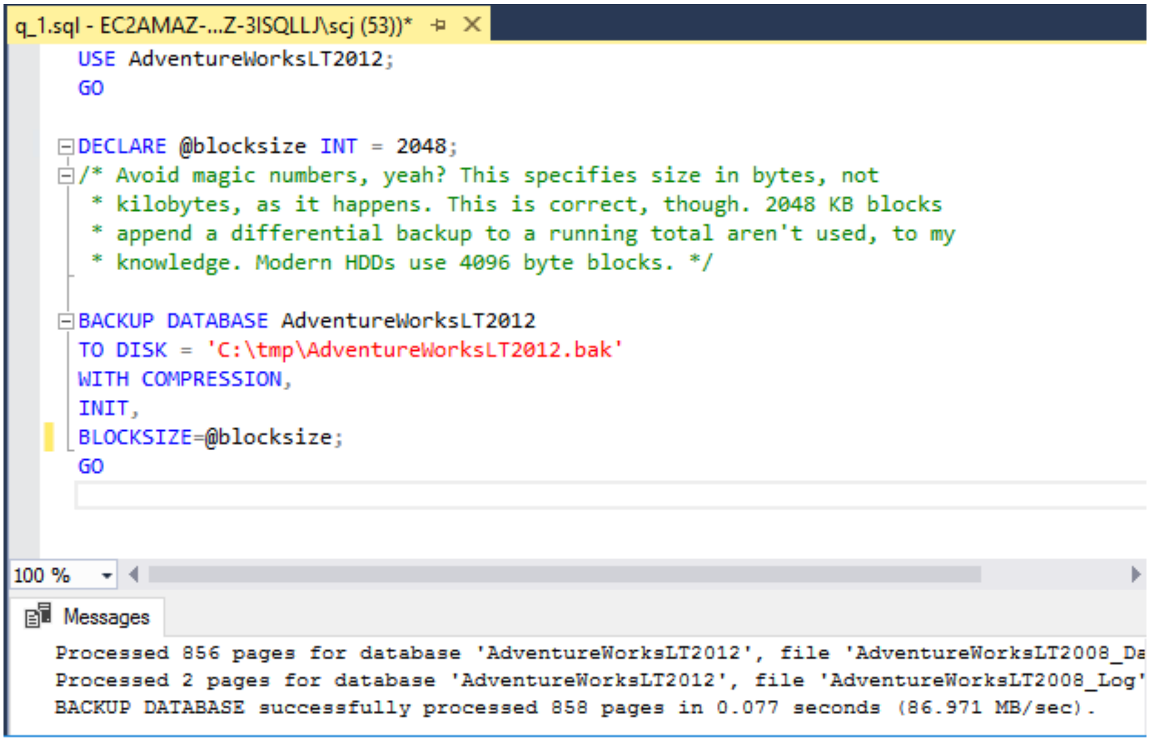
Listing 1: per 2.2

```
USE AdventureWorksLT2012;
GO

DECLARE @blocksize INT = 2048;
/* Avoid magic numbers, yeah? This specifies size in bytes, not
 * kilobytes, as it happens. This is correct, though. 2048 KB blocks
 * ~~append a differential backup to a running total~~ aren’t used, to my
 * knowledge. Modern HDDs use 4096 byte blocks. */

BACKUP DATABASE AdventureWorksLT2012
TO DISK = ’C:\tmp\AdventureWorksLT2012.bak’
WITH COMPRESSION,
INIT,
BLOCKSIZE=@blocksize;
GO
```

Figure 1: Per 1

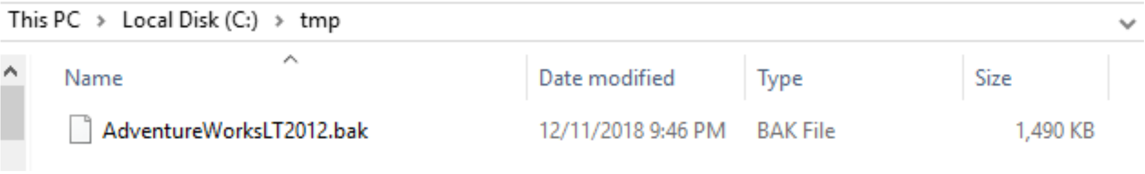


```
USE AdventureWorksLT2012;
GO

DECLARE @blocksize INT = 2048;
/* Avoid magic numbers, yeah? This specifies size in bytes, not
 * kilobytes, as it happens. This is correct, though. 2048 KB blocks
 * append a differential backup to a running total aren't used, to my
 * knowledge. Modern HDDs use 4096 byte blocks. */

BACKUP DATABASE AdventureWorksLT2012
TO DISK = 'C:\tmp\AdventureWorksLT2012.bak'
WITH COMPRESSION,
INIT,
BLOCKSIZE=@blocksize;
GO
```

```
Processed 856 pages for database 'AdventureWorksLT2012', file 'AdventureWorksLT2008_Da
Processed 2 pages for database 'AdventureWorksLT2012', file 'AdventureWorksLT2008_Log'
BACKUP DATABASE successfully processed 858 pages in 0.077 seconds (86.971 MB/sec).
```


This PC › Local Disk (C:) › tmp

Name	Date modified	Type	Size
AdventureWorksLT2012.bak	12/11/2018 9:46 PM	BAK File	1,490 KB

2.3 Backup of Six High-Load DBs

Given the availability requirements and the low number of modifications each day, instead of creating full backups each night, I would instead:

1. Create a full backup on the weekend, likely Sunday. SQL Server provides this functionality through a command.

2. Each day of the week until the next Sunday, after the high load slows down, create a differential backup. This functionality is also provided through SQL Server commands

3. Repeat this process weekly. This can be automated with the use of the SQL Server Agent component, or using some OS feature to automate this, such as cron in Linux.

2.4 Backup AdventureWorks external drive for maintenance

Figure 2: Per 2.4

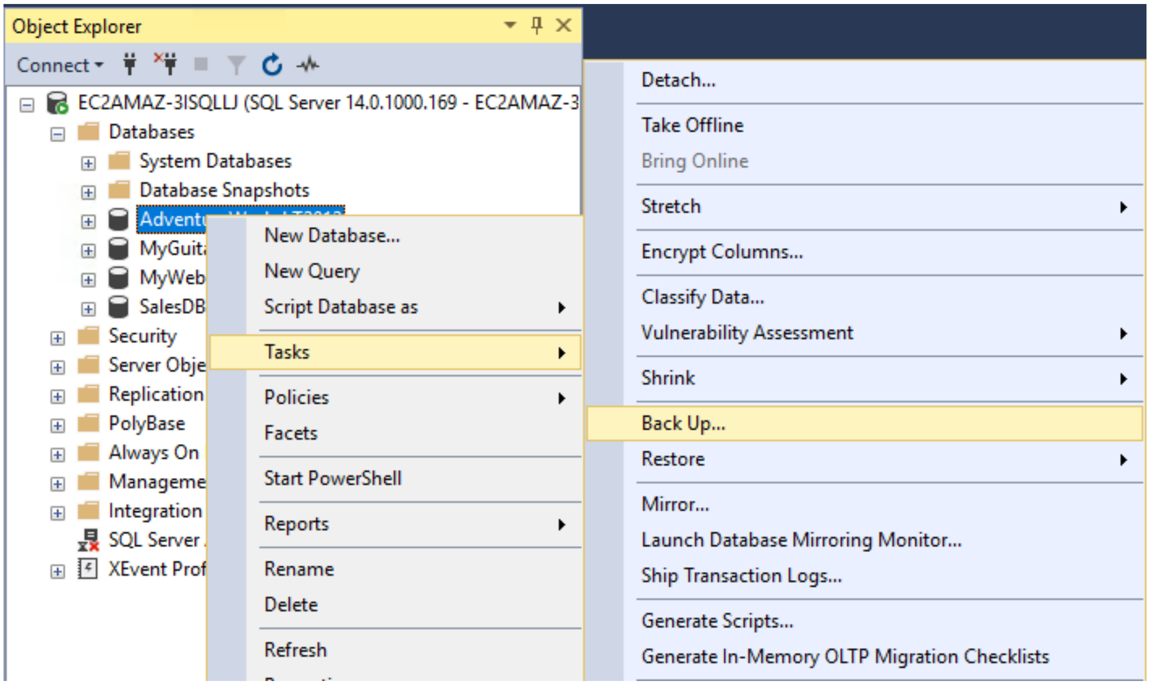
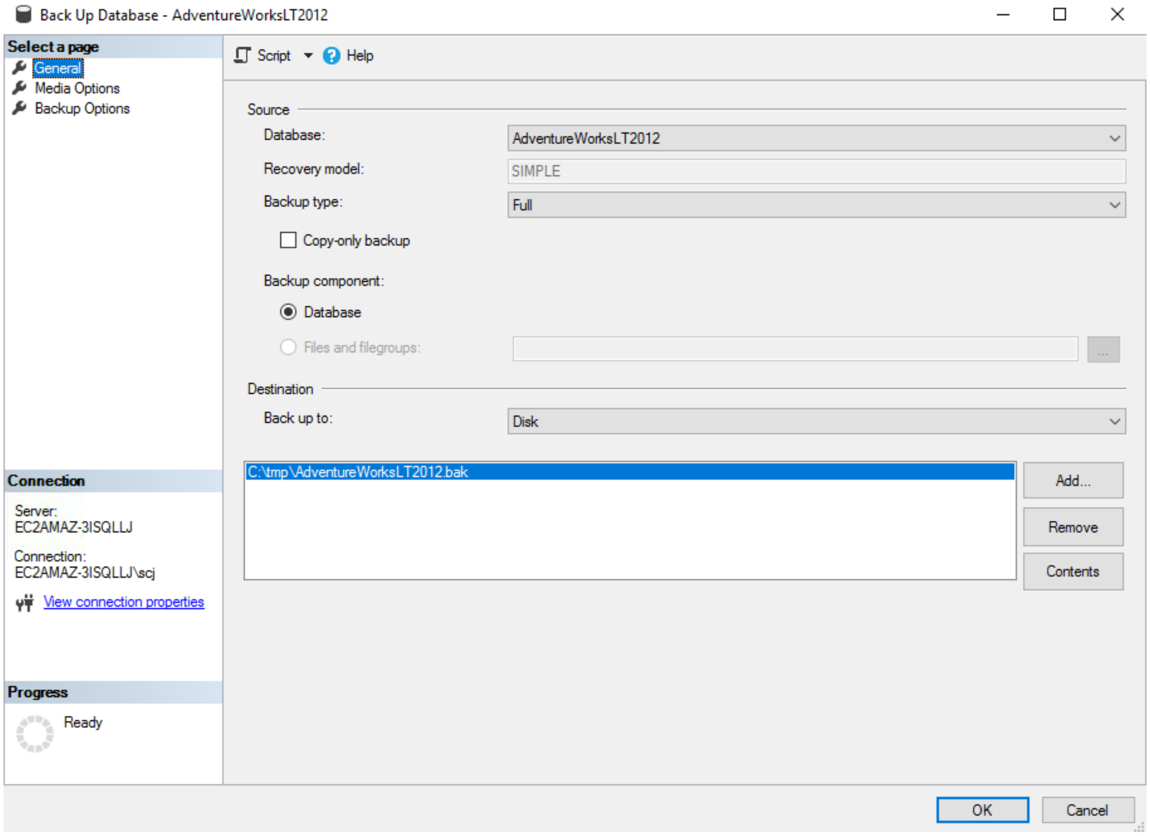


Figure 3: Per 2.4



Back Up Database - AdventureWorksLT2012

Select a page
- General
- Media Options
- Backup Options

Script ▾ Help

Source

Database: AdventureWorksLT2012

Recovery model: SIMPLE

Backup type: Full

□ Copy-only backup

Backup component:

● Database

○ Files and filegroups:

Destination

Back up to: Disk

C:\tmp\AdventureWorksLT2012.bak

Connection

Server:
EC2AMAZ-3ISQLLJ

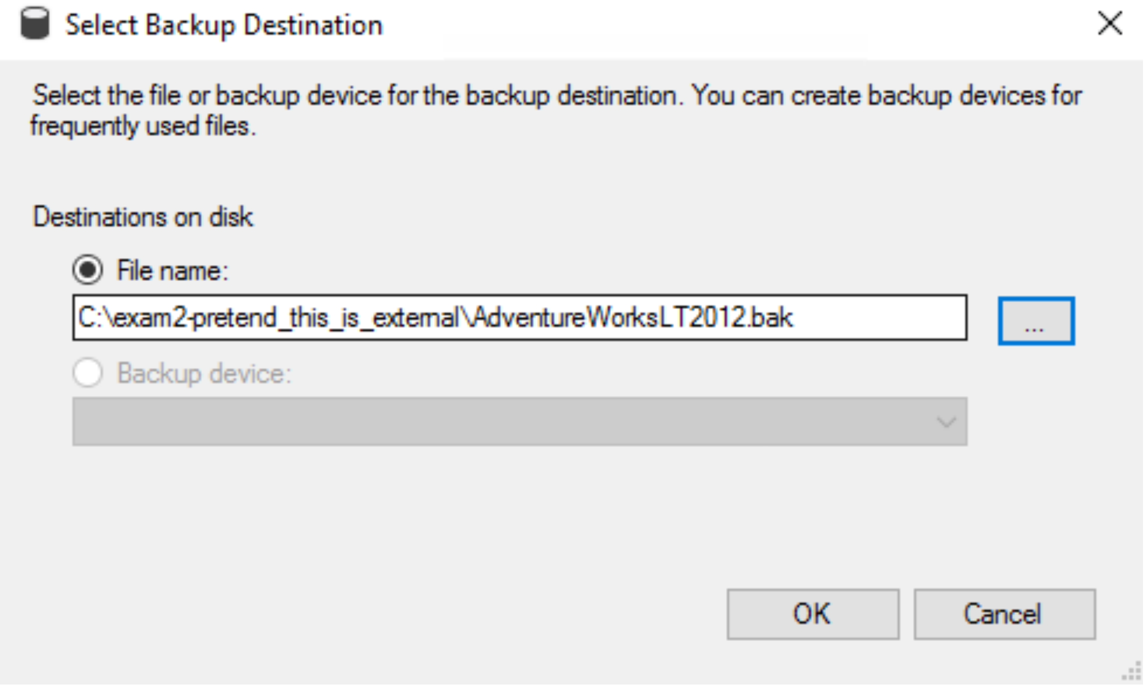
Connection:
EC2AMAZ-3ISQLLJ\scj

View connection properties

Progress

Ready

Figure 4: Per 2.4



Select Backup Destination

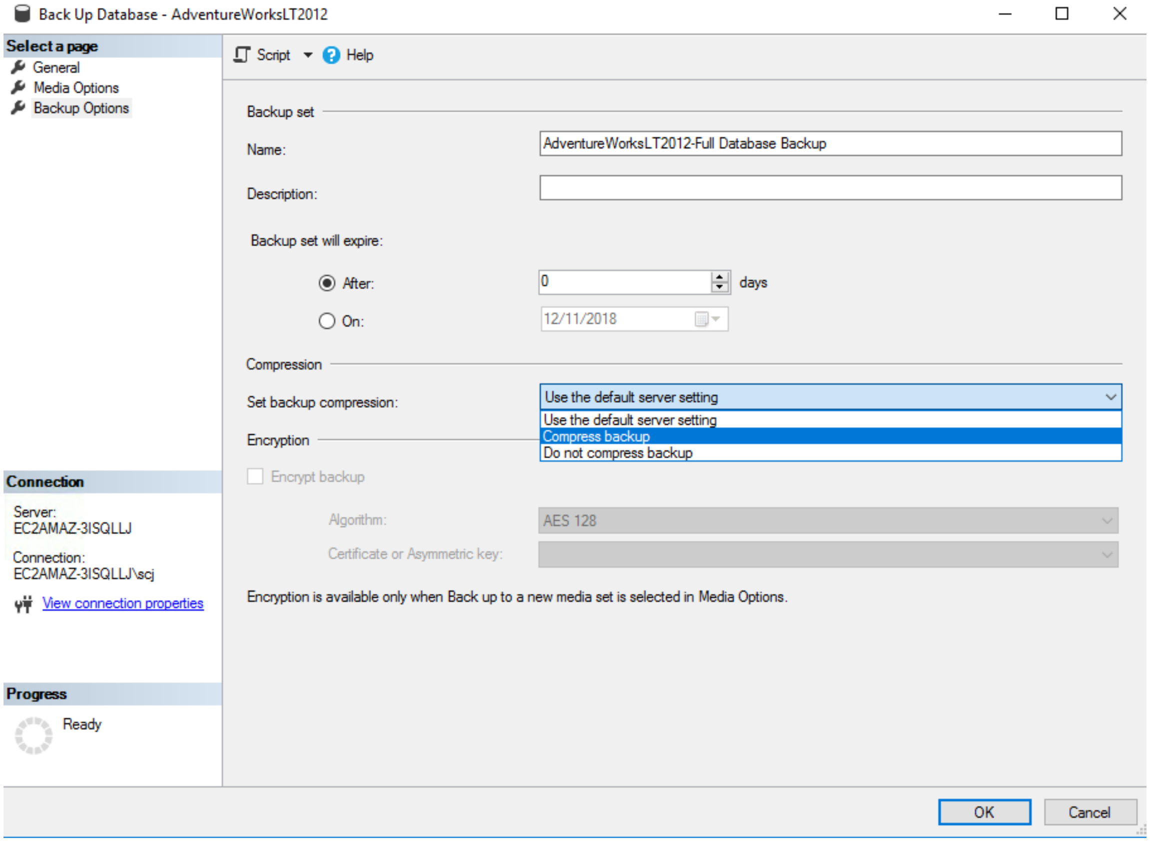
Select the file or backup device for the backup destination. You can create backup devices for frequently used files.

Destinations on disk

● File name:

C:\exam2-pretend_this_is_external\AdventureWorksLT2012.bak

Figure 5: Per 2.4



Back Up Database - AdventureWorksLT2012

Backup set

Name: AdventureWorksLT2012-Full Database Backup

Description:

Backup set will expire:

● After: 0 days

○ On: 12/11/2018

Compression

Set backup compression: Use the default server setting

Encryption

□ Encrypt backup

Algorithm: AES 128

Certificate or Asymmetric key:

Encryption is available only when Back up to a new media set is selected in Media Options.

Connection

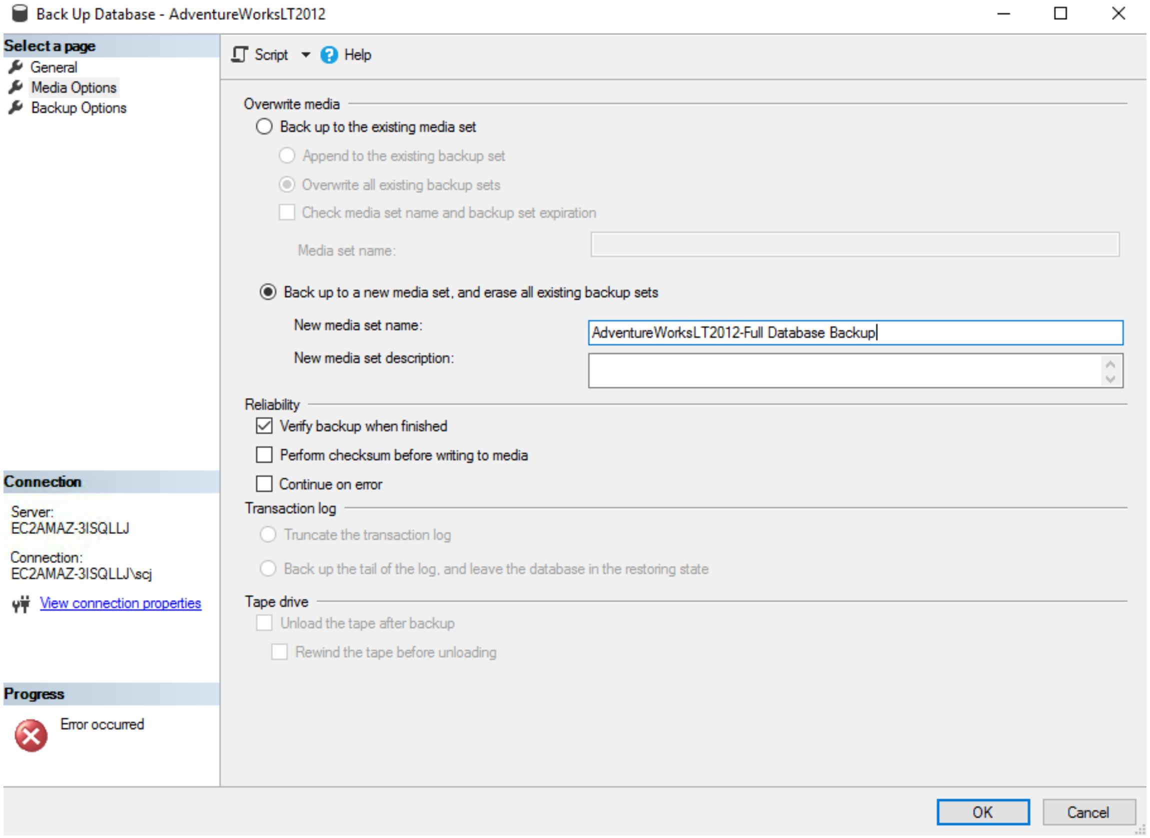
Server:
EC2AMAZ-3ISQLLJ

Connection:
EC2AMAZ-3ISQLLJ\scj

Progress

Ready

Figure 6: Per 2.4



Back Up Database - AdventureWorksLT2012

Select a page
- General
- Media Options
- Backup Options

Script ▾ Help

Overwrite media

○ Back up to the existing media set

○ Append to the existing backup set

● Overwrite all existing backup sets

□ Check media set name and backup set expiration

Media set name:

● Back up to a new media set, and erase all existing backup sets

New media set name: AdventureWorksLT2012-Full Database Backup

New media set description:

Reliability

☑ Verify backup when finished

□ Perform checksum before writing to media

□ Continue on error

Transaction log

○ Truncate the transaction log

○ Back up the tail of the log, and leave the database in the restoring state

Tape drive

□ Unload the tape after backup

□ Rewind the tape before unloading

Connection

Server:
EC2AMAZ-3ISQLLJ

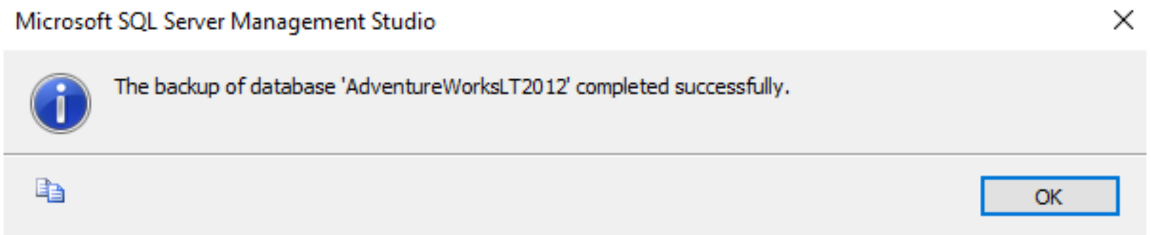
Connection:
EC2AMAZ-3ISQLLJ\scj

View connection properties

Progress

Error occurred

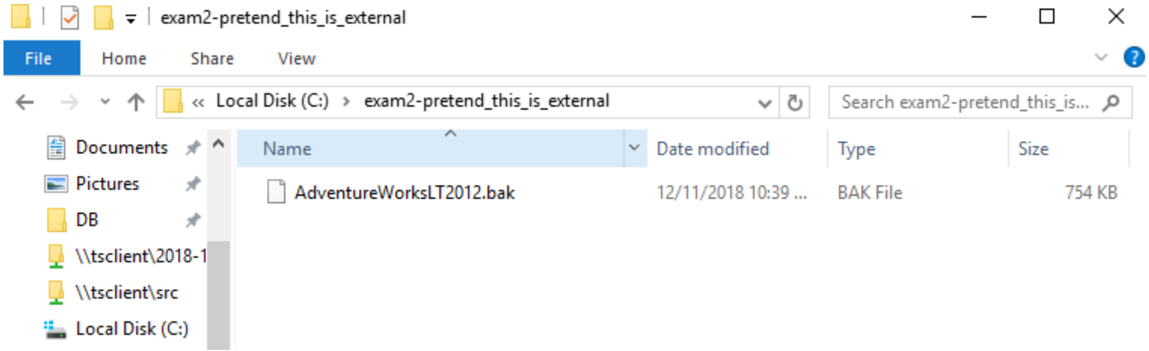
Figure 7: Per 2.4



Microsoft SQL Server Management Studio

The backup of database 'AdventureWorksLT2012' completed successfully.

Figure 8: Per 2.4


exam2-pretend_this_is_external
File
Home
Share
View
Local Disk (C:) › exam2-pretend_this_is_external
Search exam2-pretend_this_is...
Documents
Pictures
DB
\\tsclient\2018-1
\\tsclient\src
Local Disk (C:)

Name	Date modified	Type	Size
AdventureWorksLT2012.bak	12/11/2018 10:39 ...	BAK File	754 KB

2.5 Recover AdventureWorks from backup

Figure 9: Per 2.5

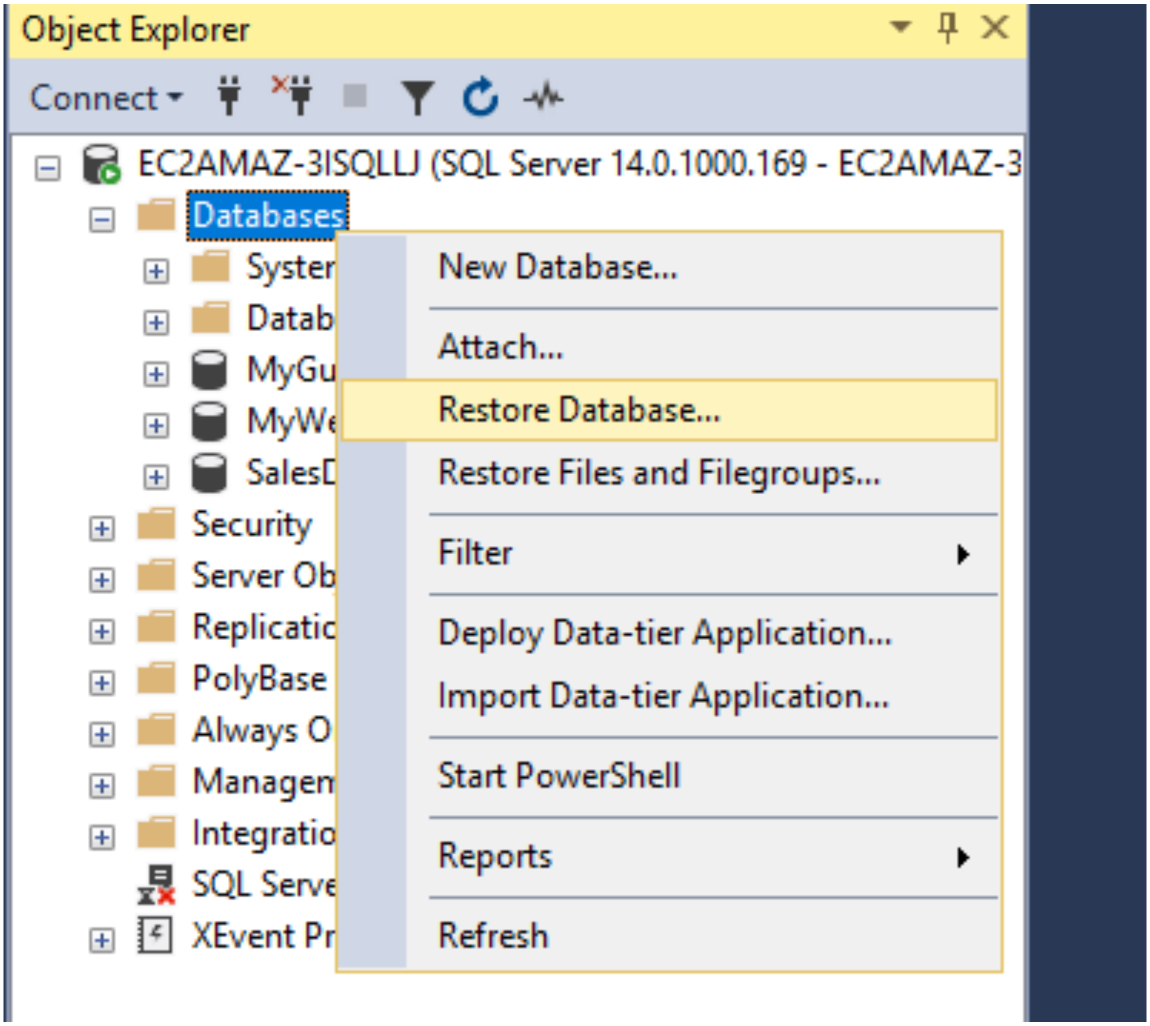
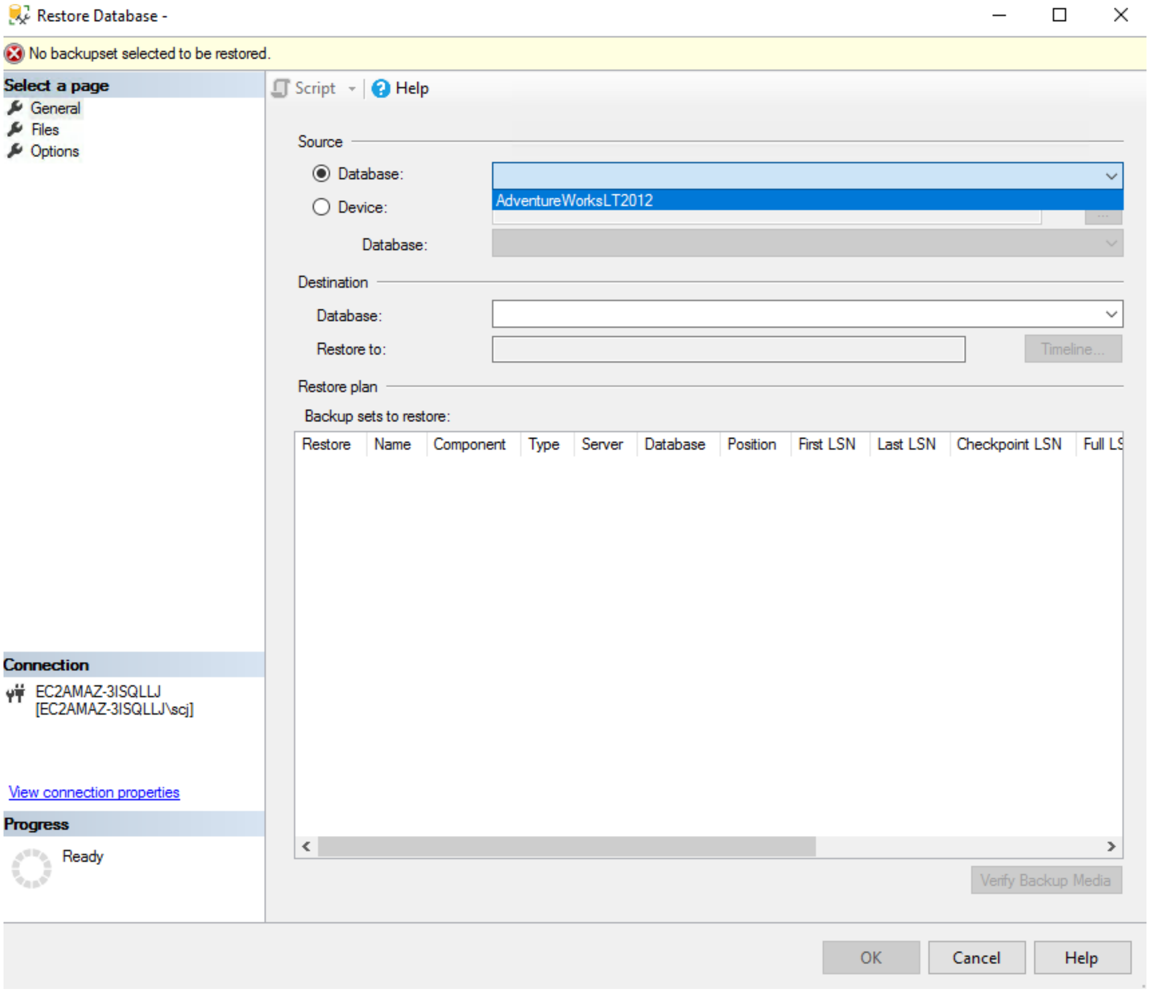


Figure 10: Per 2.5



Restore Database -

No backupset selected to be restored.

Source

● Database: AdventureWorksLT2012

○ Device:

Database:

Destination

Database:

Restore to:

Restore plan

Backup sets to restore:

Restore	Name	Component	Type	Server	Database	Position	First LSN	Last LSN	Checkpoint LSN	Full LS

Connection

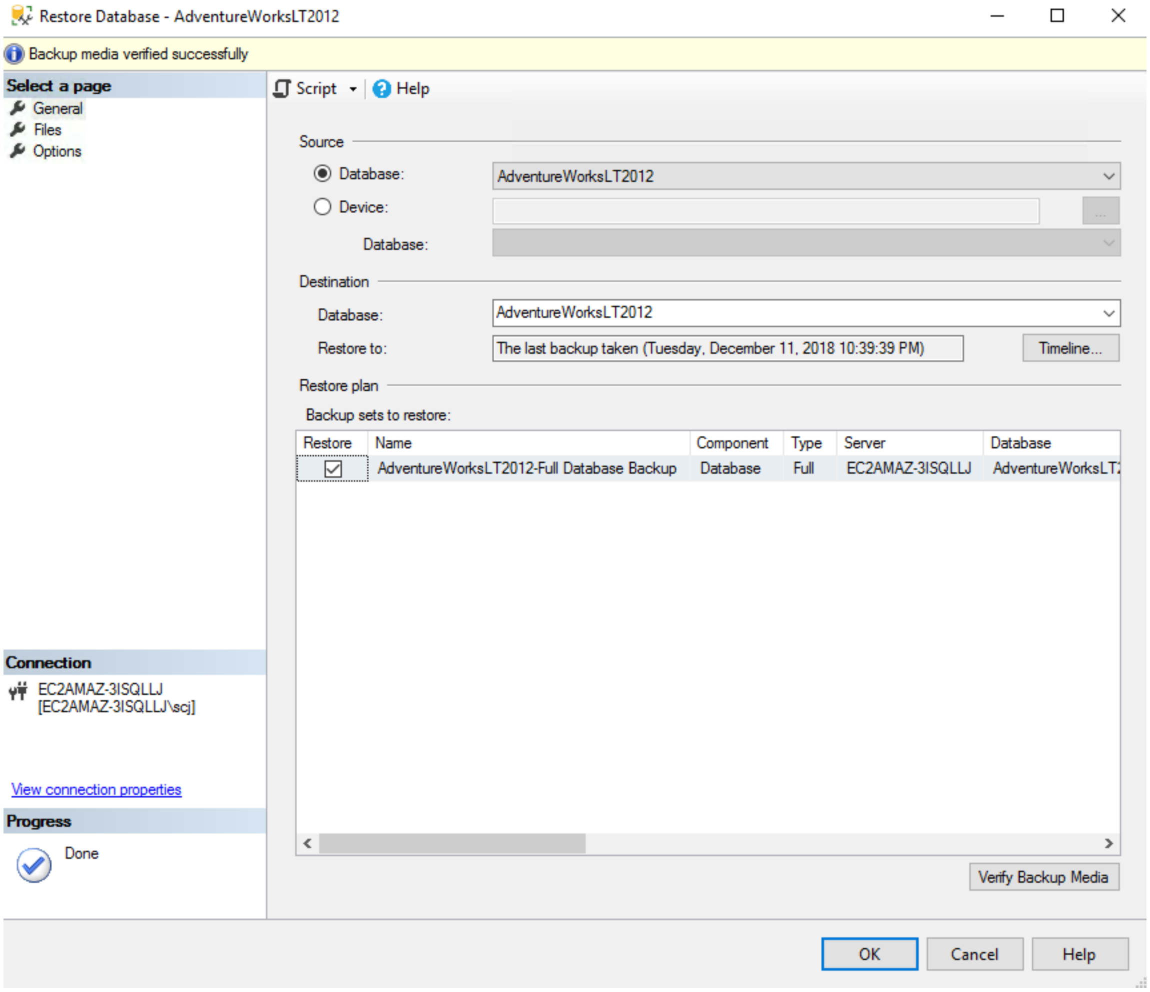
EC2AMAZ-3ISQLLJ
[EC2AMAZ-3ISQLLJ\scj]

Progress

Ready

Figure 11: Per 2.5

Figure 12: Per 2.5



Restore Database - AdventureWorksLT2012

Backup media verified successfully

Source

● Database: AdventureWorksLT2012

○ Device:

Database:

Destination

Database: AdventureWorksLT2012

Restore to: The last backup taken (Tuesday, December 11, 2018 10:39:39 PM)

Restore plan

Backup sets to restore:

Restore	Name	Component	Type	Server	Database
☑	AdventureWorksLT2012-Full Database Backup	Database	Full	EC2AMAZ-3ISQLLJ	AdventureWorksLT2

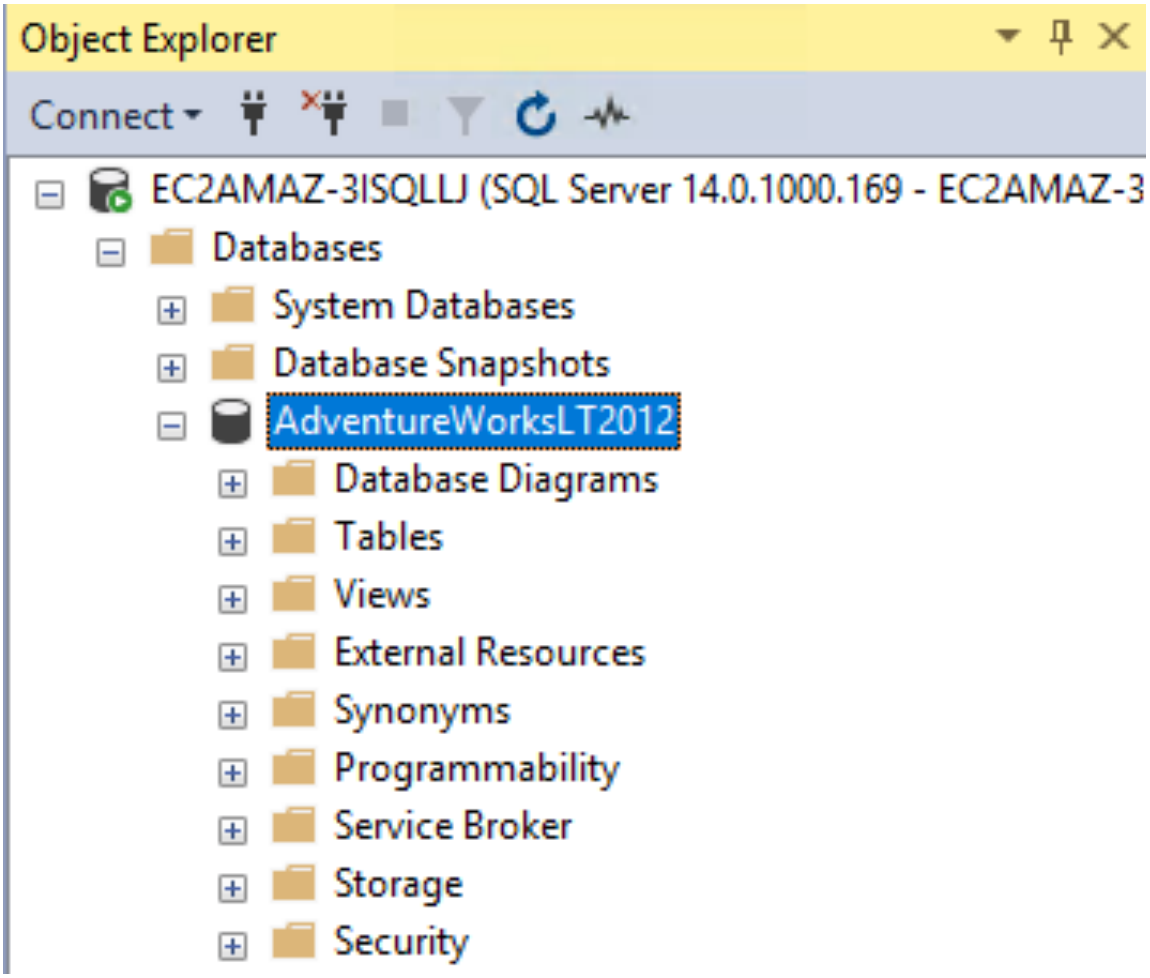
Connection

EC2AMAZ-3ISQLLJ
[EC2AMAZ-3ISQLLJ\scj]

Progress

Done

Figure 13: Per 2.5



2.6 Disaster Recovery site in the cloud

Best practices for backups of any kind are to have multiple physical locations where data is stored, both an on-site backup and an off-site backup. Contracting another nearby company to handle physically redundant backups would be a lengthy and costly process, not to mention the installation of equipment to make a space a competent warm or hot site. Then of course there are physical security needs to be handled. Using one of a number of secure cloud hosts, such as through Amazon's Web Services or Microsoft's cloud platform, we can ensure the confidentiality, integrity, and availability of our data and operations at relatively low cost.

Cloud service providers generally have very scalable pricing models, such that unused space and time is not charged for. There is of course still a charge for idle use and disk storage, but this would be much lower than the idle cost of a physically redundant site. Configuration is fast, inexpensive, and easy for most common technology stacks. The only real bottleneck by comparison is the time to send data over the wire to the cloud service provider's servers we rent.

Hosting a DR site in the cloud several other benefits not limited to:

- Far-flung remote sites are unlikely to be affected by power outages or dangerous weather we may experience locally.
- Duplicating our DR site across availability zones is trivial.
- The service provider already has security procedures in place, both physical and digital.