SQL Security

The Value of Data

- Some say it's the organization's **most important** asset.
- Avoid these problems to protect your data.
  - **Data inconsistency**—Multiple data names and definitions for the same data or inconsistent representation of the same data living in many places within an organization
  - **Missing data**—Data that should exist but doesn't because the organization doesn't value it
  - **Poor quality**—Data is entered into the database with appropriate data hygiene
  - **Poor timing**—Data that is too old to be of use
  - **Poor data awareness**—The organization isn't aware of the data that already exists
  - **Poor availability**—It takes too long to get the data when it's needed caused by outages, excessive controls, and poor response time
  - **Poor controls**—The data is misused, is misrepresented, or is used by the unauthorized creating opportunities for fraud and breaches in privacy and security

Protecting Data

- **Access control**—Use of a DBMS's authentication capabilities for validating trusted users and applications. Use of the database's authorization capabilities to control access to facilities and data applicable to an employee's job duties. We will explore access control in this week's lab.
- **Integrity control**—Use of a DBMS's integrity constraints: entity integrity, referential integrity, check, assertion, and rule. (We've already learned how to do this.)
- **Detection control**—Use of the DBMS's facilities to detect and track intruders through the use of logging, system, and table triggers
- **Obscurity control**—Use of a DBMS's facilities to "hide" data that include extensive used of the external data model with views and stored procedures, encryption, and virtual private databases (row-level security).
- **Resource control**—Use of the DBMS's facilities that control resource consumption like CPU time, elapsed time, memory, and disk I/Os. Not all DBMSs have these capabilities.
- **Redundancy control**—Use of the DBMS's backup/recovery facilities that include hot and cold backups, logging, journaling, checkpoints, archivin,g and mirroring. We'll talk more about this next.

Database Security

Use SQL commands to control access to various objects.

Securable Object	Permissions
Database	BACKUP DATABASE, BACKUP LOG, CREATE DATABASE, CREATE Objects
Scalar function	EXECUTE
Stored procedure	DELETE, EXECUTE, INSERT, SELECT, and UPDATE
Table	DELETE, INSERT, REFERENCES, SELECT, and UPDATE
View	SELECT

Database Security Implemented

- Creating a database user (example)

```
CREATE USER supply_chain FROM LOGIN
supply_chain
```

Database Security Implemented

- Granting permission (example)

```
GRANT EXECUTE ON ApplyMarkup to
username
```

- Revoking (denying) permission to objects

```
REVOKE INSERT, UPDATE, DELETE ON
Customer TO username
```

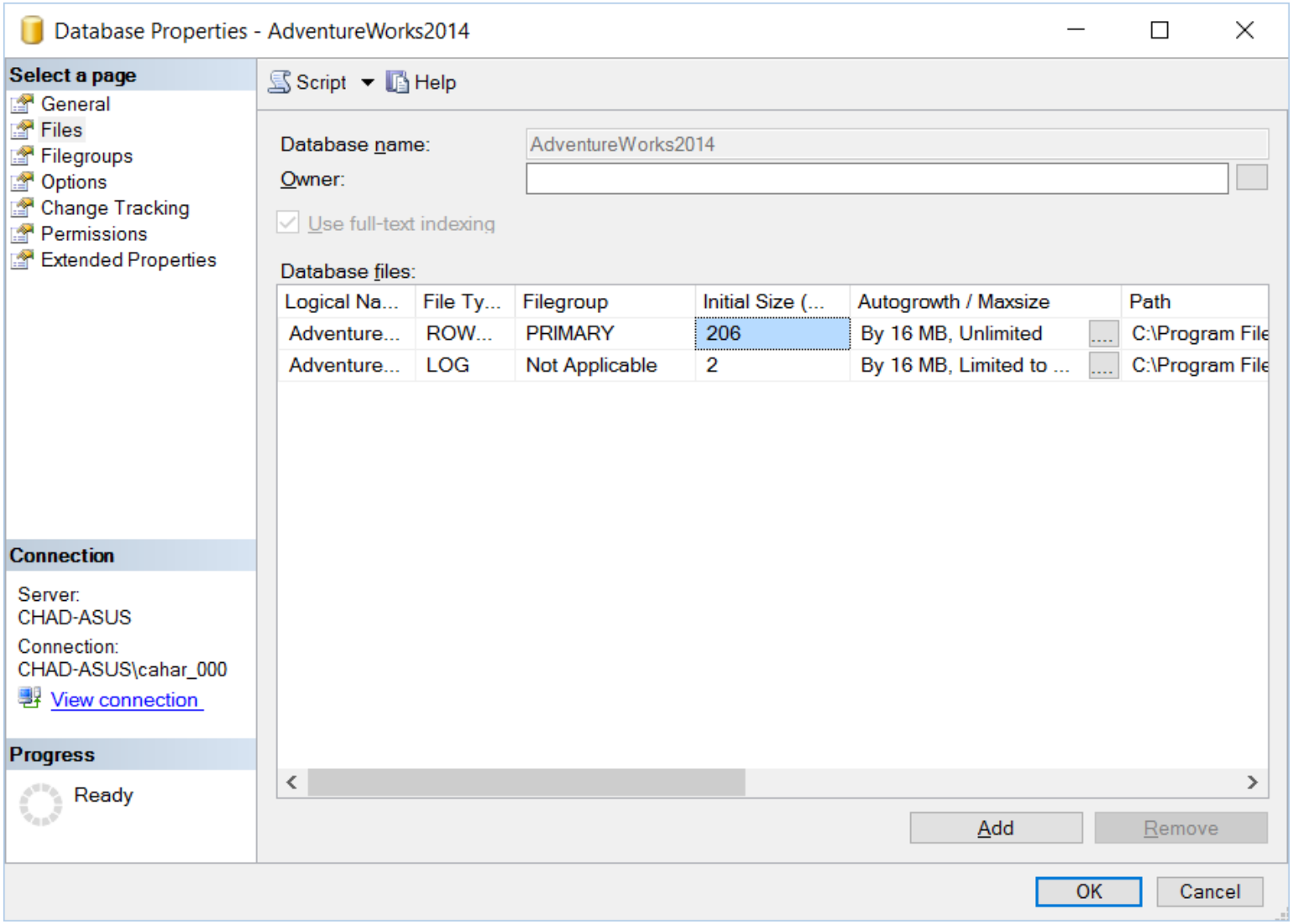
Backup and Restore

Simple Model

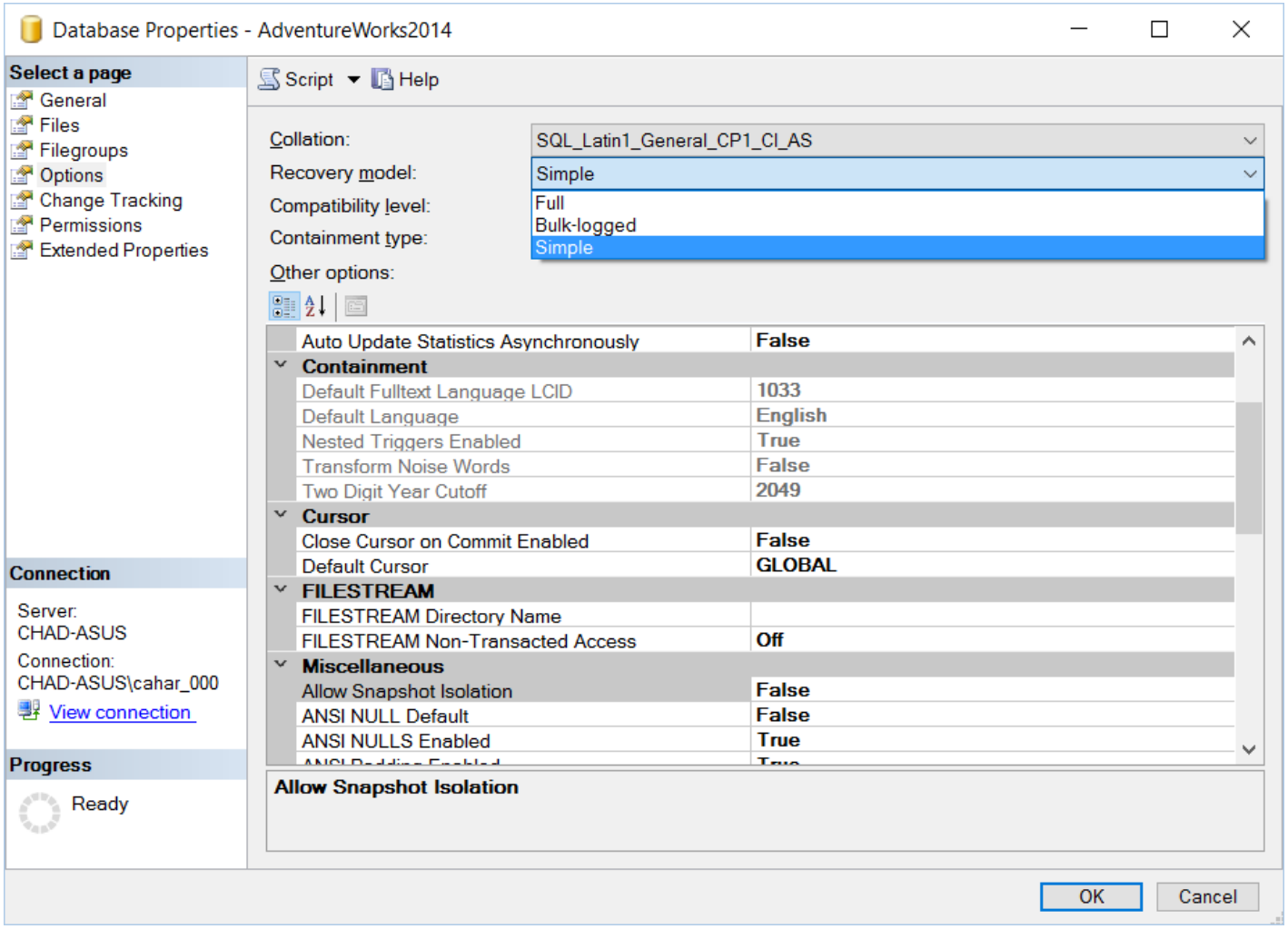
Simple Model

- Restore or back up the database in its entirety at the point in time of the backup.
- Useful when
  - Data changes infrequently
  - Database is a test environment
  - Data can be easily recreated

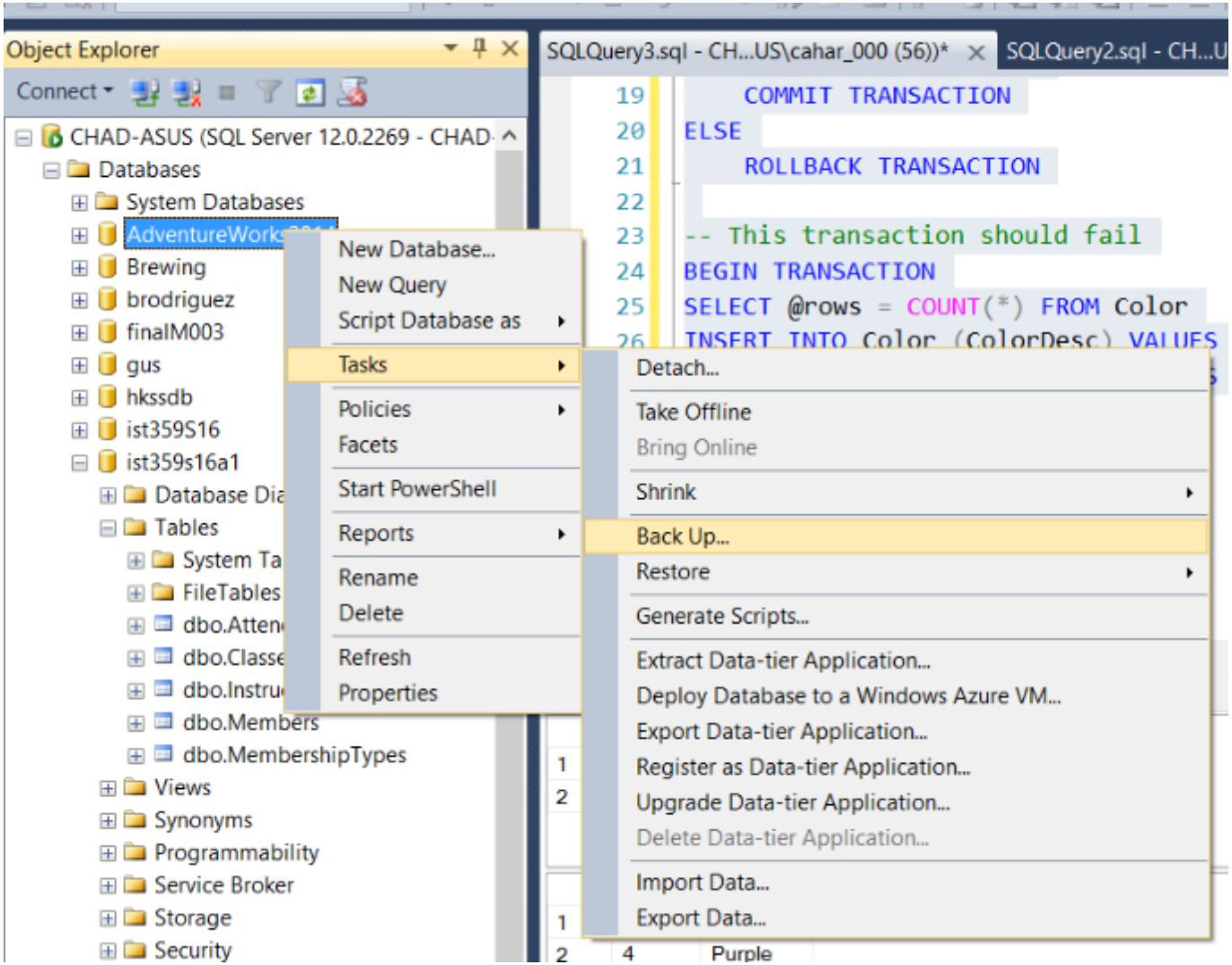
Simple Model



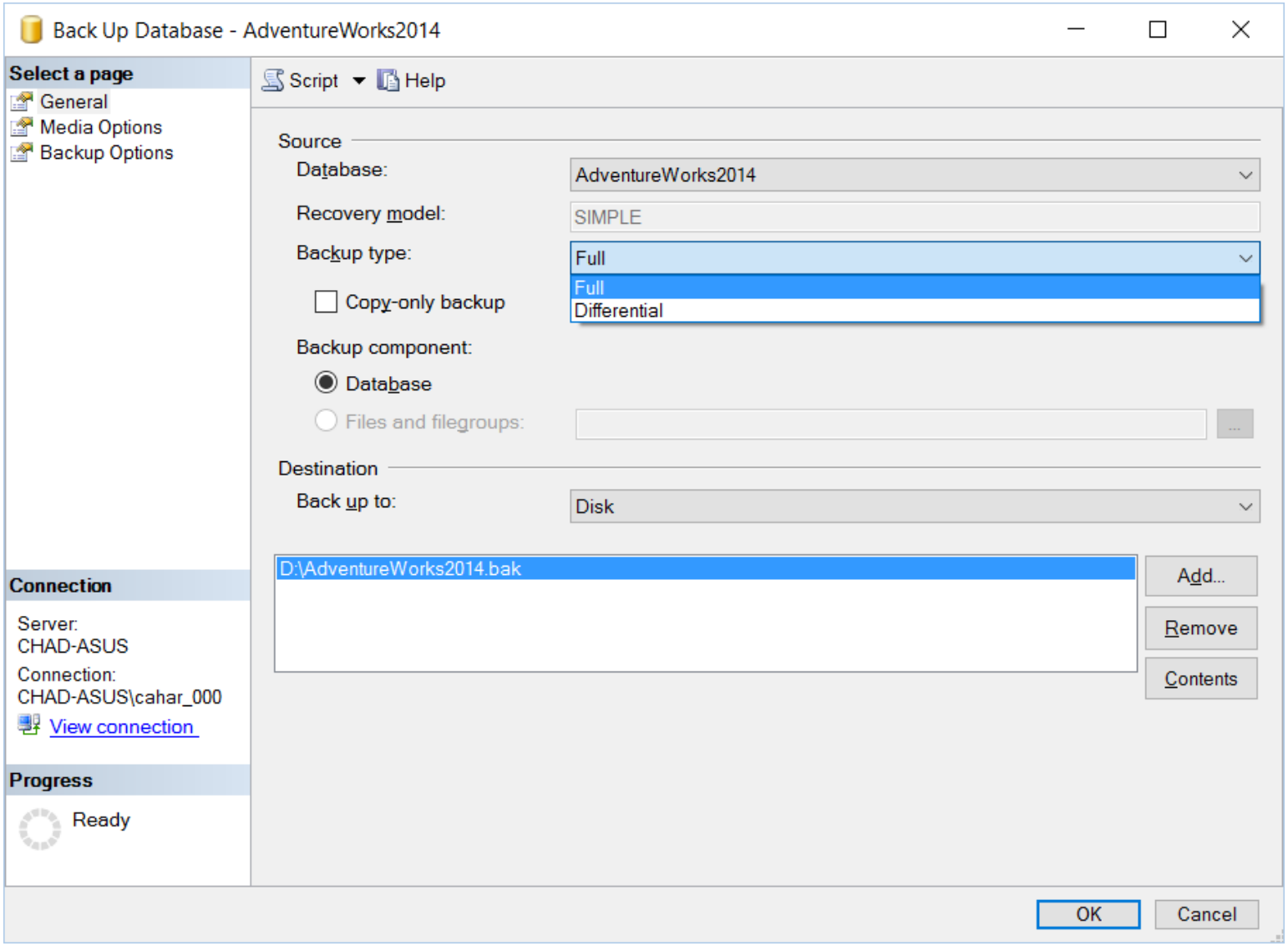
Simple Model



Simple Model



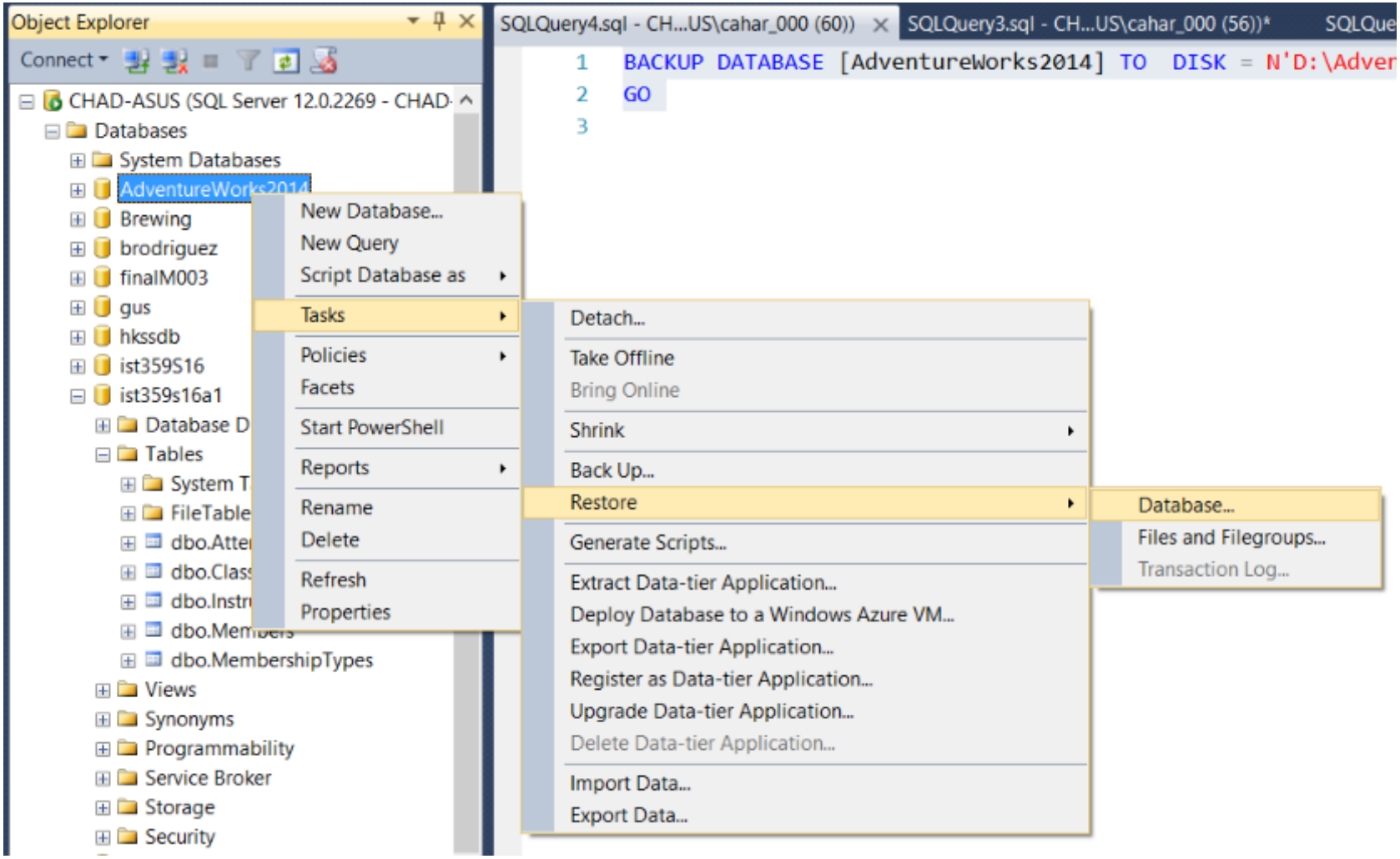
Simple Model



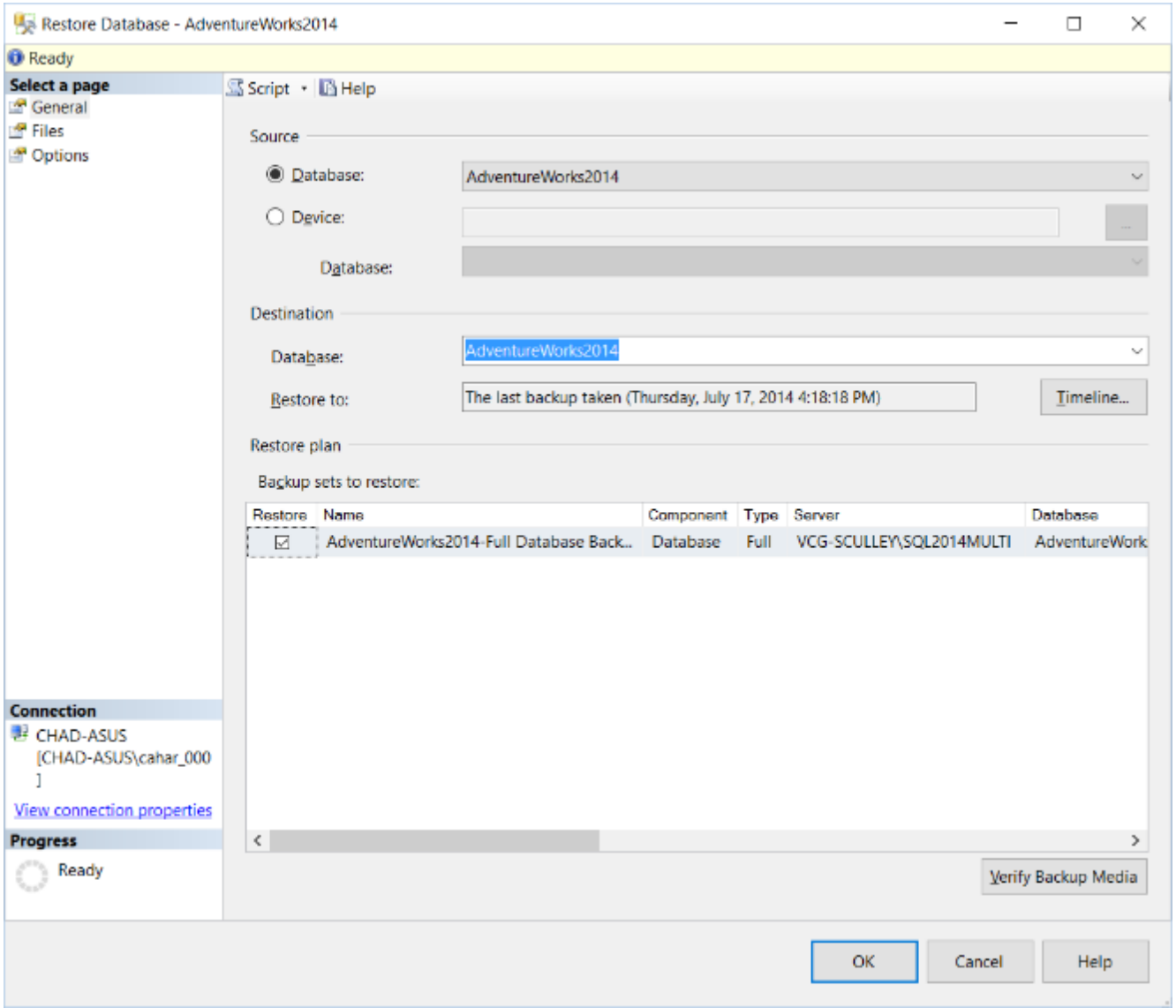
Simple Model (in SQL)

```
BACKUP DATABASE [AdventureWorks2014]
TO  DISK = N'D:\AdventureWorks2014.bak'
WITH NAME = N'AdventureWorks2014-Full Database Backup'
GO
```

Simple Model (Recovery)



Simple Model (Recovery)



Simple Model (Recovery)

```
USE [master]
RESTORE DATABASE [AdventureWorks2014]
FROM DISK = N'D:\AdventureWorks2014.bak'
WITH  FILE = 1
GO
```

Backup and Restore

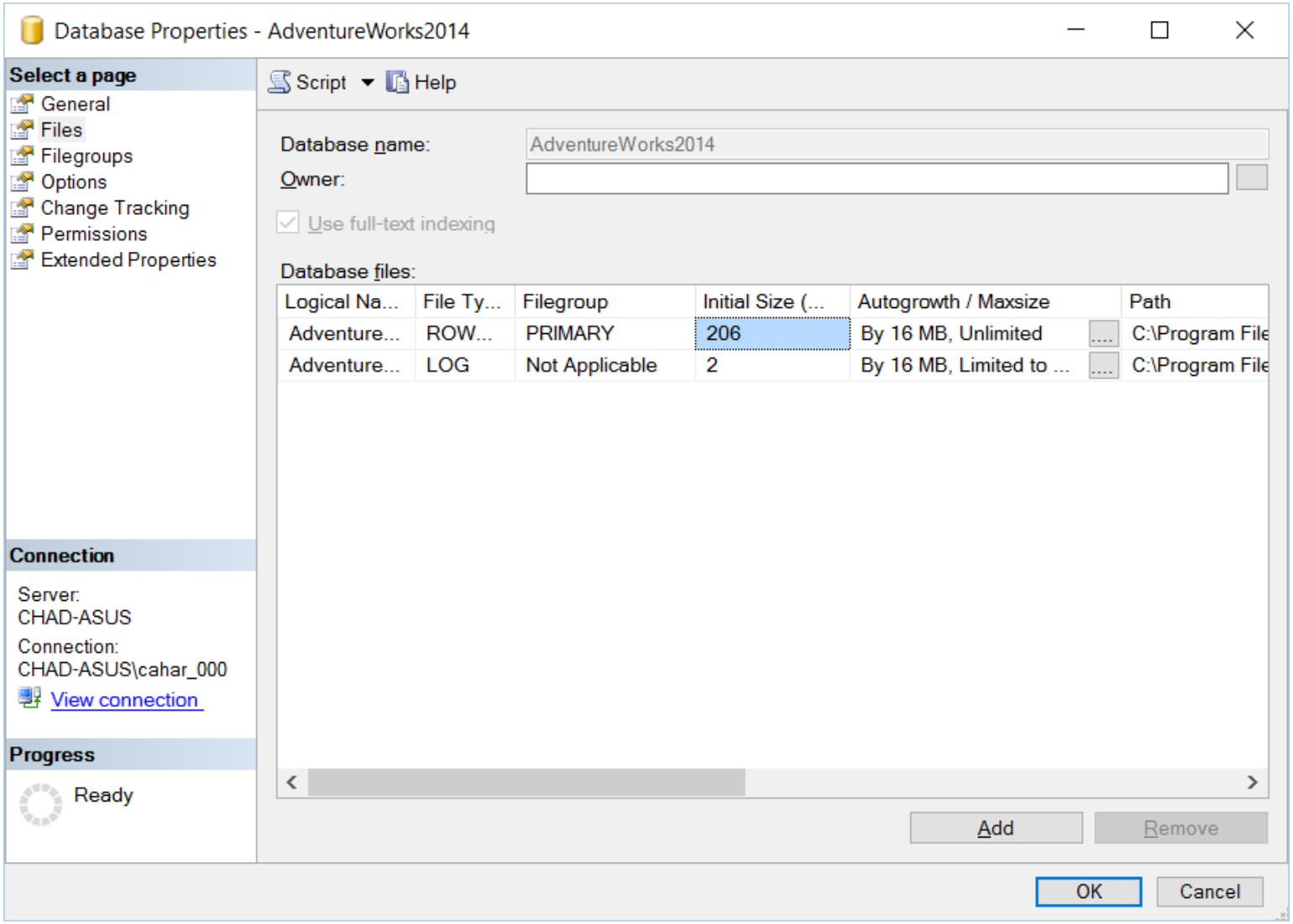
Full Model

Full Model

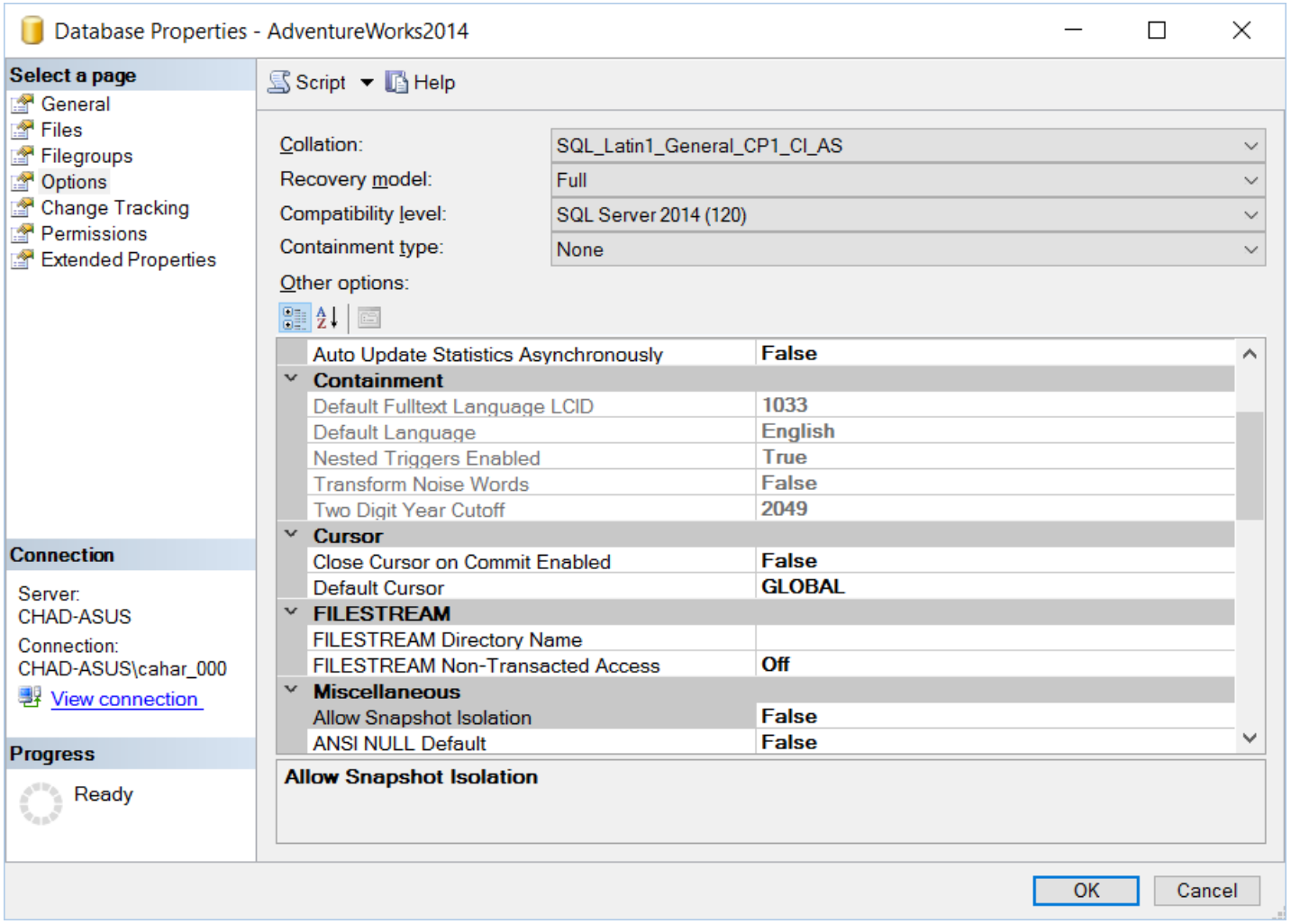
- Can restore a backup to a particular point in time (including the state right before the restore)

- Useful when
  - Data are constantly changing
  - Database is in a production environment with continual activity
  - Data are not easily recreated

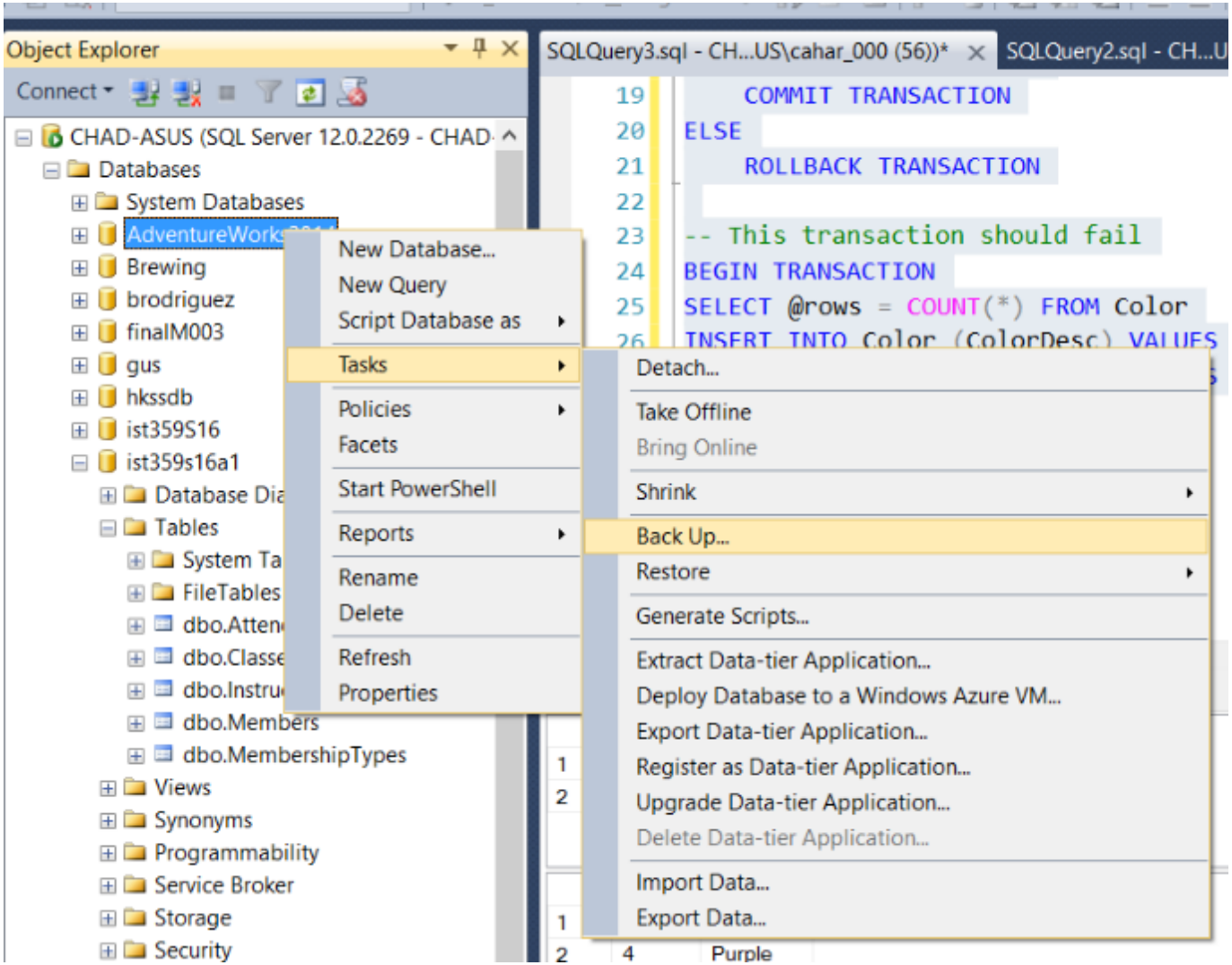
Full Model


Database Properties - AdventureWorks2014
Select a page
General
Files
Filegroups
Options
Change Tracking
Permissions
Extended Properties
Script
Help
Database name:
AdventureWorks2014
Owner:
Use full-text indexing
Database files:
Logical Na...
File Ty...
Filegroup
Initial Size (...
Autogrowth / Maxsize
Path
Adventure...
ROW...
PRIMARY
206
By 16 MB, Unlimited
C:\Program File
Adventure...
LOG
Not Applicable
2
By 16 MB, Limited to ...
C:\Program File
Add
Remove
Connection
Server:
CHAD-ASUS
Connection:
CHAD-ASUS\cahar_000
View connection
Progress
Ready
OK
Cancel

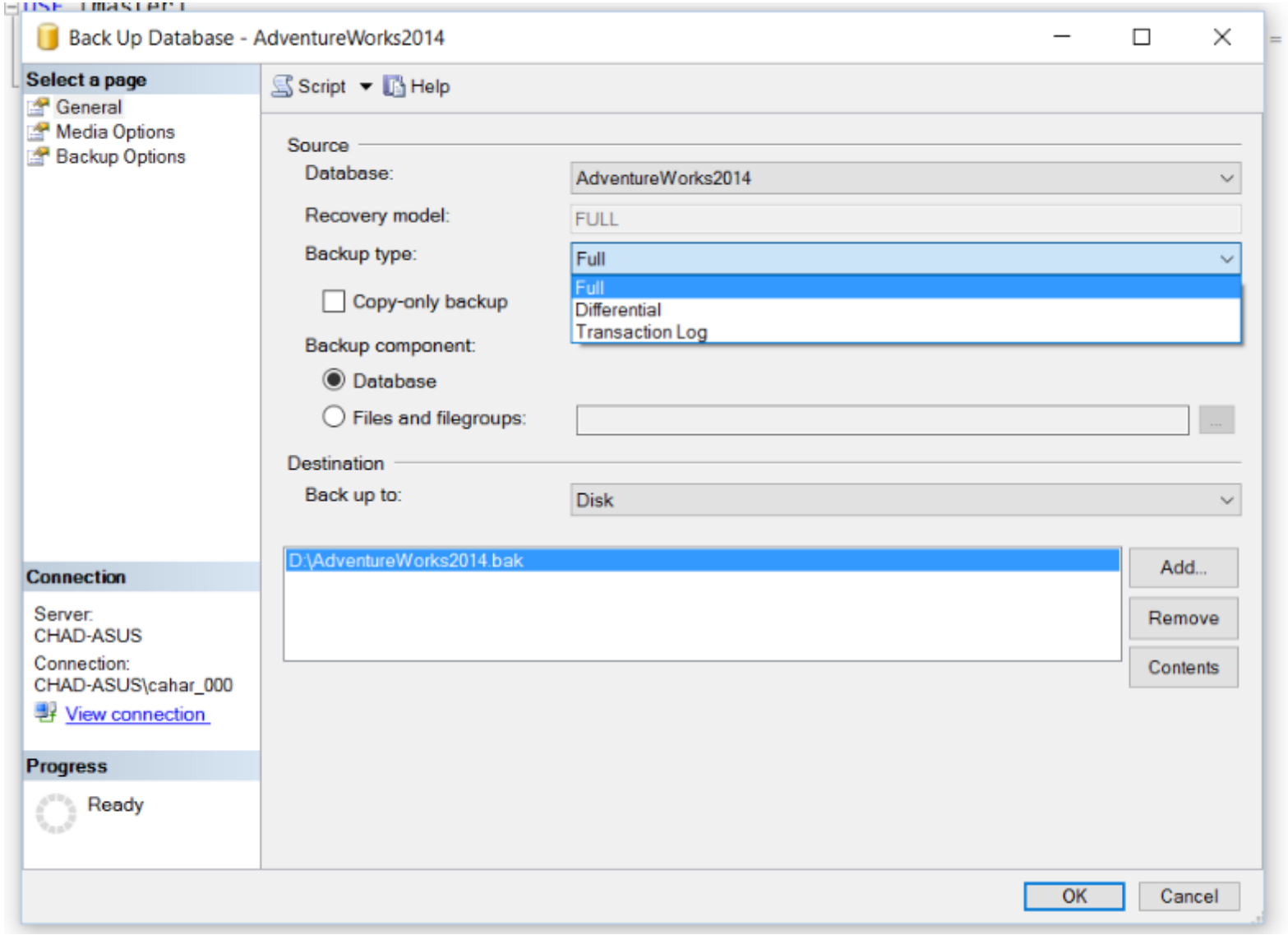
Full Model


Database Properties - AdventureWorks2014
Select a page
General
Files
Filegroups
Options
Change Tracking
Permissions
Extended Properties
Script
Help
Collation:
SQL_Latin1_General_CP1_CI_AS
Recovery model:
Full
Compatibility level:
SQL Server 2014 (120)
Containment type:
None
Other options:
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:
CHAD-ASUS
Connection:
CHAD-ASUS\cahar_000
View connection
Progress
Ready
OK
Cancel

Full Model


Object Explorer
Connect
CHAD-ASUS (SQL Server 12.0.2269 - CHAD
Databases
System Databases
AdventureWorks
Brewing
brodriguez
finalM003
gus
hkssdb
ist359S16
ist359s16a1
Database Dia
Tables
System Ta
FileTables
dbo.Atten
dbo.Classe
dbo.Instru
dbo.Members
dbo.MembershipTypes
Views
Synonyms
Programmability
Service Broker
Storage
Security
New Database...
New Query
Script Database as
Tasks
Policies
Facets
Start PowerShell
Reports
Rename
Delete
Refresh
Properties
Detach...
Take Offline
Bring Online
Shrink
Back Up...
Restore
Generate Scripts...
Extract Data-tier Application...
Deploy Database to a Windows Azure VM...
Export Data-tier Application...
Register as Data-tier Application...
Upgrade Data-tier Application...
Delete Data-tier Application...
Import Data...
Export Data...
SQLQuery3.sql - CH...US\cahar_000 (56))*
SQLQuery2.sql - CH...U
19 COMMIT TRANSACTION
20 ELSE
21 ROLLBACK TRANSACTION
22
23 -- This transaction should fail
24 BEGIN TRANSACTION
25 SELECT @rows = COUNT(*) FROM Color
26 INSERT INTO Color (ColorDesc) VALUES
1
2
1
2 4 Purple

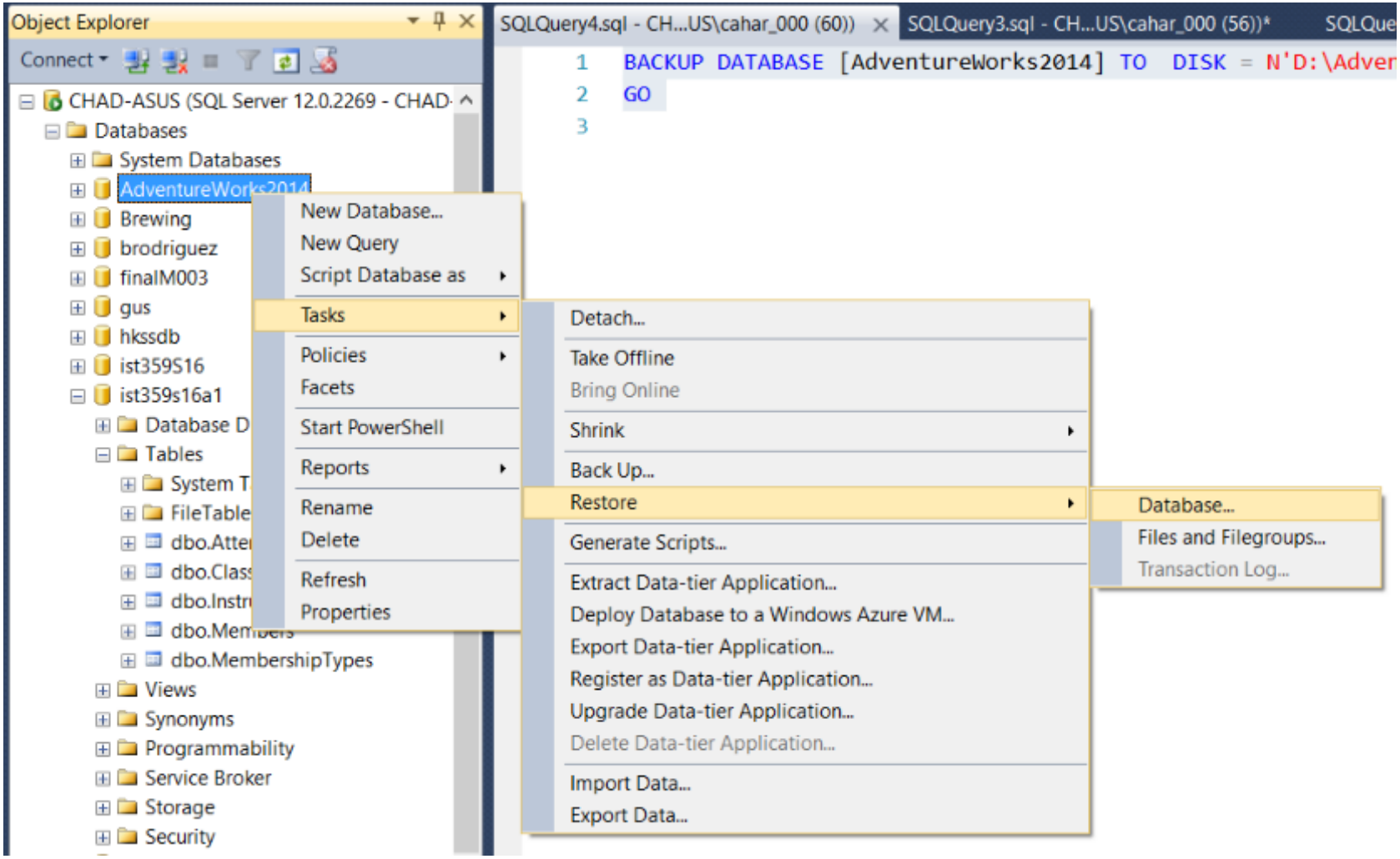
Full Model


Back Up Database - AdventureWorks2014
Select a page
General
Media Options
Backup Options
Script
Help
Source
Database:
AdventureWorks2014
Recovery model:
FULL
Backup type:
Full
Full
Differential
Transaction Log
Copy-only backup
Backup component:
Database
Files and filegroups:
Destination
Back up to:
Disk
D:\AdventureWorks2014.bak
Add...
Remove
Contents
Connection
Server:
CHAD-ASUS
Connection:
CHAD-ASUS\cahar_000
View connection
Progress
Ready
OK
Cancel

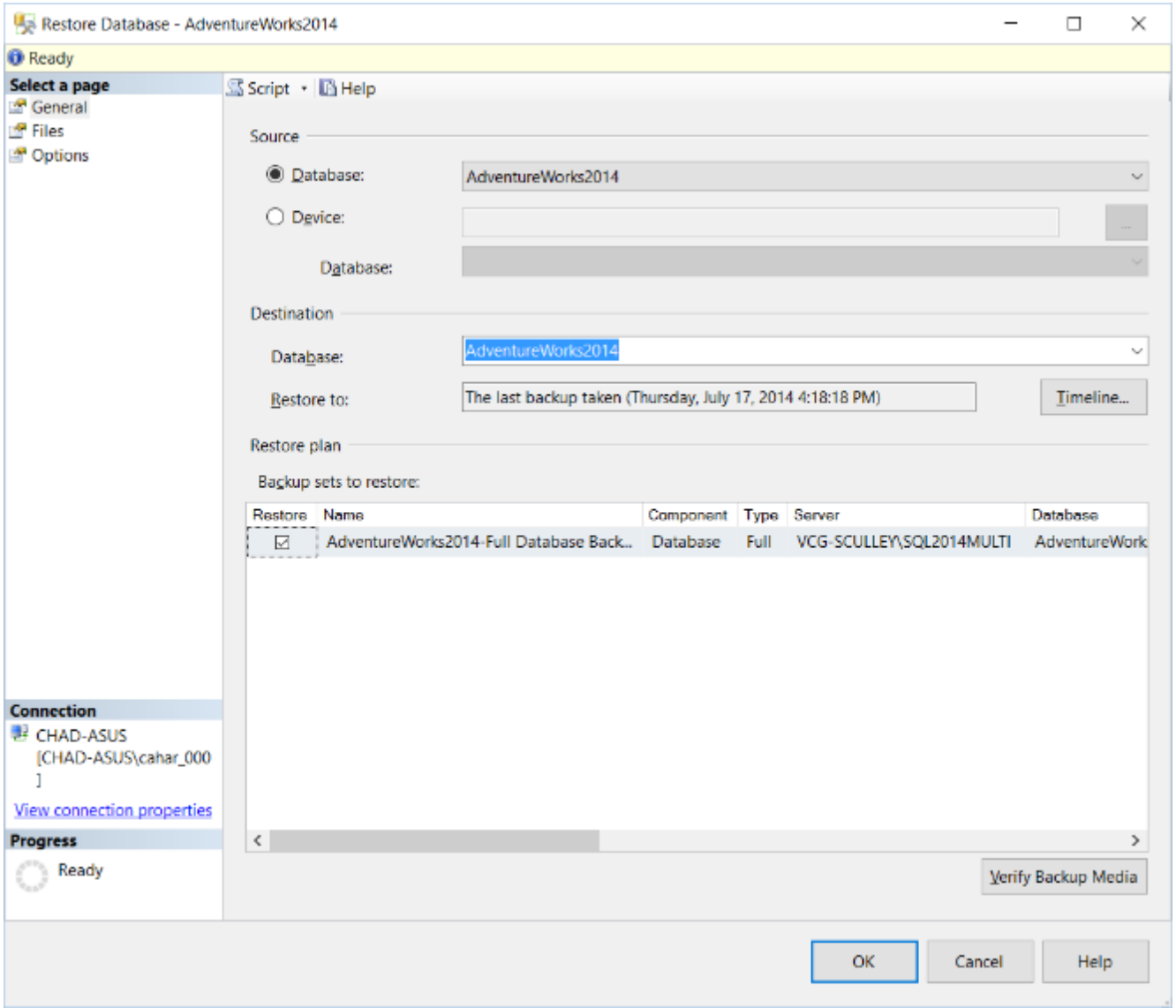
Full Model (SQL)

```
BACKUP (LOG|DATABASE)
[AdventureWorks2014]
TO  DISK = N'D:
\AdventureWorks2014.bak‘ [WITH
(DIFFERENTIAL|
NAME = N'AdventureWorks2014-Full
Database Backup‘
GO
```

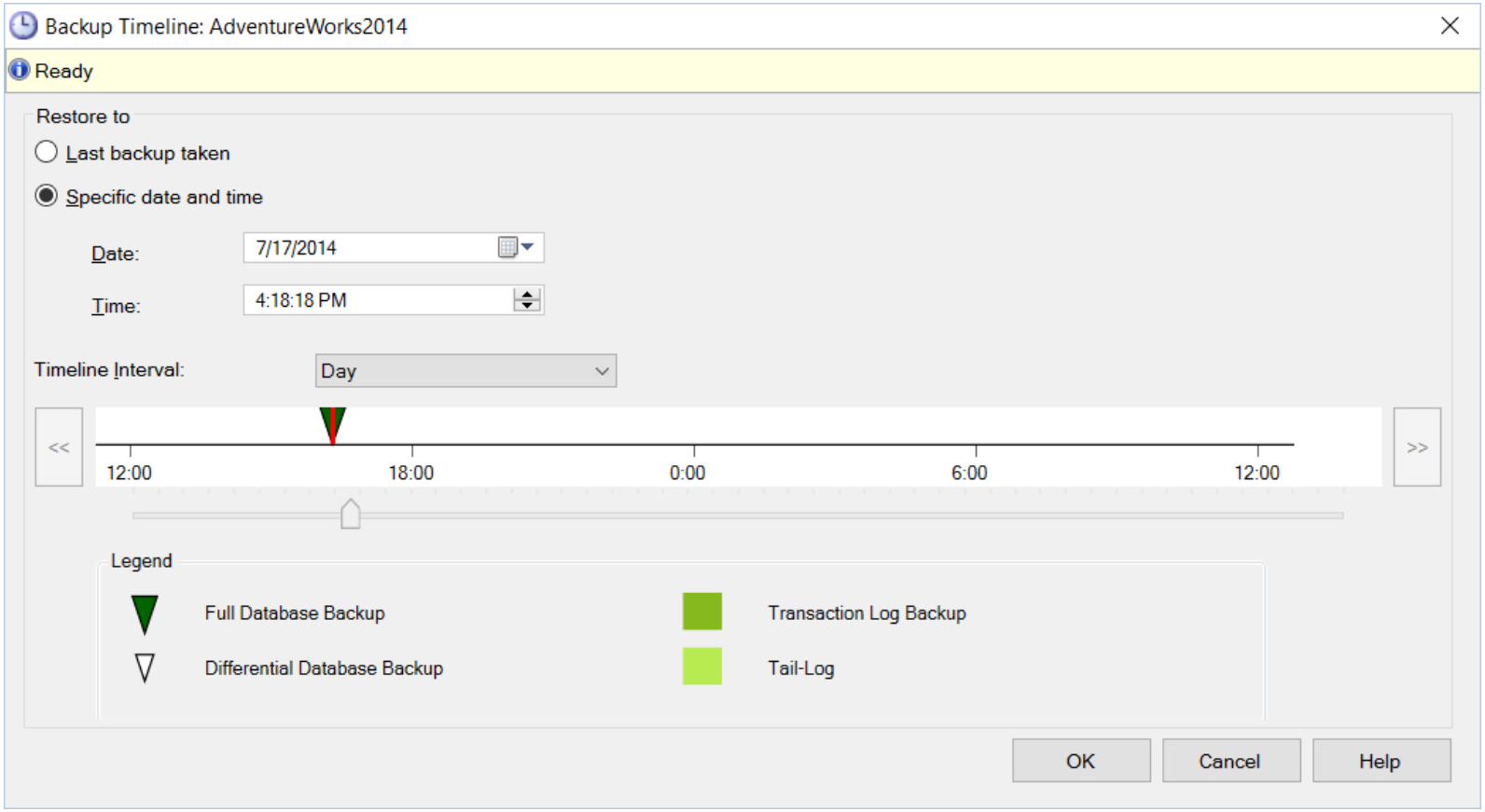
Full Model (Recovery)



Full Model (Recovery)



Full Model (Recovery)



Database Performance and Tuning

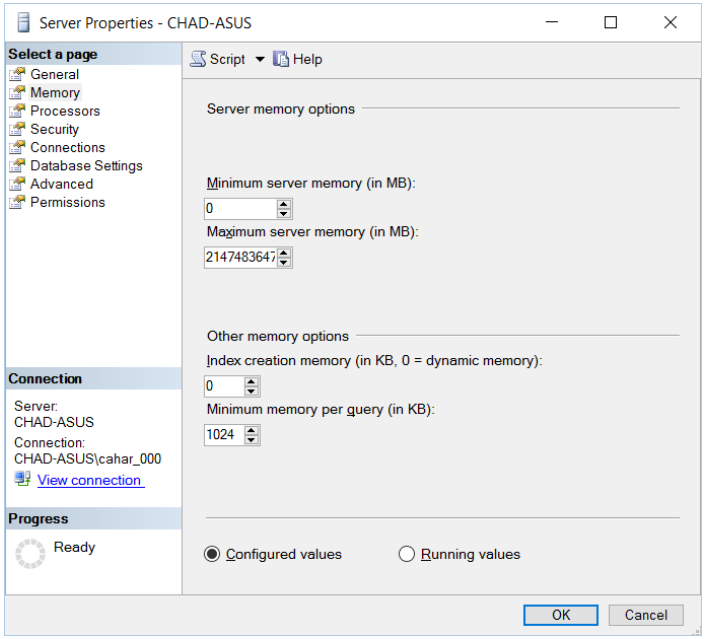
Database Performance

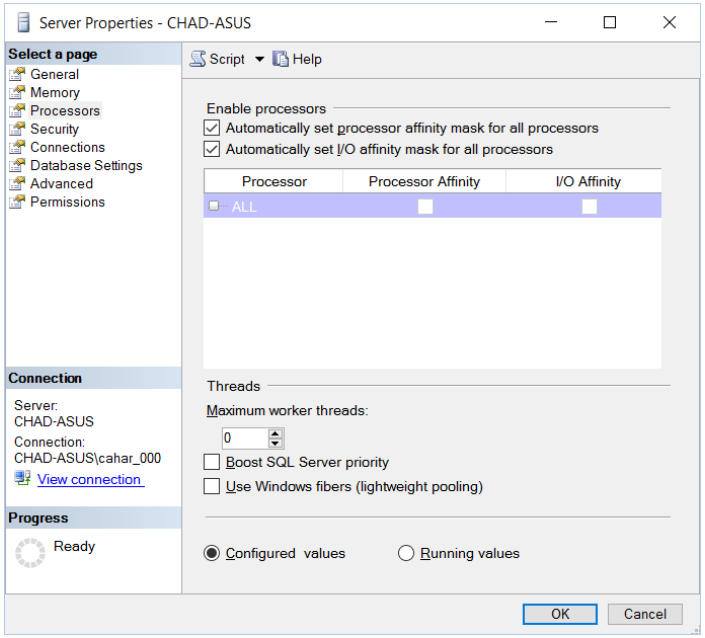
Common hindrances to optimal performance

- Hardware and networking
- Underperforming queries
- Mismanaged indexes

Database Performance

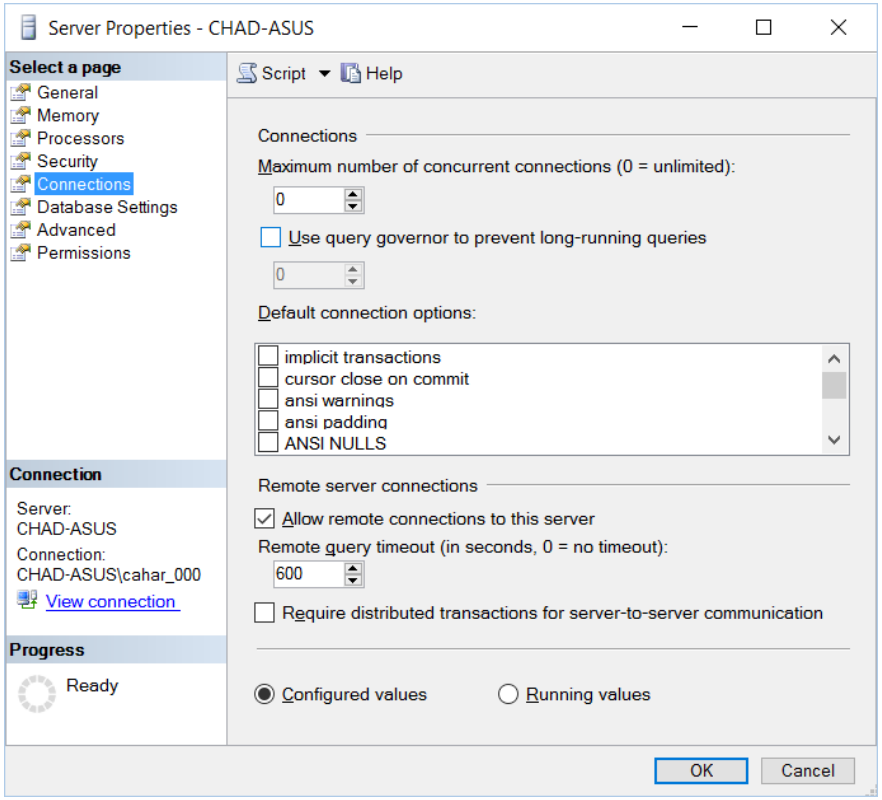
Hardware and networking





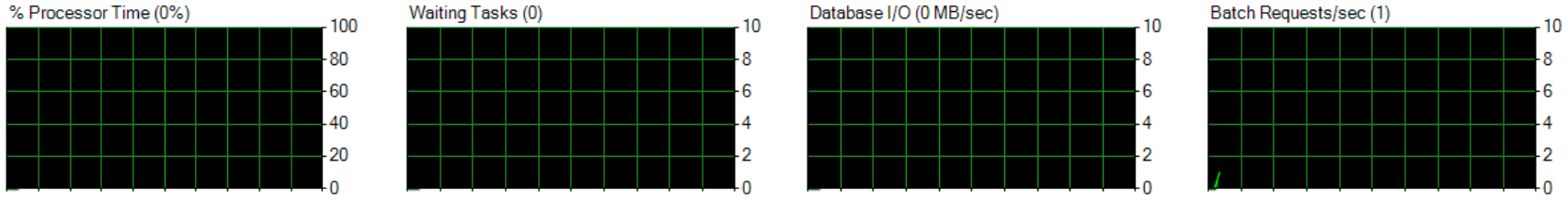
Database Performance

Hardware and networking



CHAD-ASUS - Activity Monitor | SQLQuery11.sql - C...US\cahar_000 (58)) | SQLQuery10.sql - C...US\cahar_000 (57)) | SQLQuery9.sql - CH...US\cahar_000 (55)) | SQLQuery8.sql - C

Overview



Processes

S	U	L...	D...	T...	C...	A...	Wait ...	W...	W...	B...	H...	M...	H...	W...
51	1	chad-...	tempdb	RUNN...	SELE...	Micro...	0					24	CHAD...	internal
52	1	chad-...	master			Micro...	0					16	CHAD...	internal
53	1	chad-...	master			Micro...	0					24	CHAD...	internal
55	1	chad-...	master			Micro...	0					16	CHAD...	internal
57	1	chad-...	Adven...			Micro...	0					16	CHAD...	internal
58	1	chad-...	Adven...			Micro...	0					16	CHAD...	internal

Resource Waits

Wait Category	Wait Time (ms/sec)	Recent Wait Time...	Average Waiter C...	Cumulative Wait Tim...
Buffer I/O	0	34	0.0	13
Buffer Latch	0	0	0.0	0
Latch	0	0	0.0	0
Lock	0	0	0.0	2
Logging	0	0	0.0	0
Memory	0	0	0.0	0
Network I/O	0	0	0.0	0

Database Performance

- Underperforming queries
  - Holding locks for too long
  - Holding locks when they don't need them
  - Use of inefficient SQL constructs
  - Consider adding the `nolock` hint after each table in the from clause of SELECT statements, e.g.:
    ```
    SELECT * FROM Person NOLOCK
    ```
  - Use the SQL Profiler tool to check the performance of queries
  - Backup and flush your transaction log regularly
  - Choose appropriate data types for your fields

Database Performance

- Mismanaged indexes
  - Consider table partitioning for extremely large tables.
  - Use SQL Profiler tool to identify which indexes are being used and make sure they're well managed.
  - Use the Database Engine Tuning Advisor to evaluate the workload on the server.

Concurrency Control

Concurrency Control

- Concurrency relates to more than one process attempting to access and/or manipulate a resource.

- Layers to think about:
  - The using process's memory space
  - The server's memory space
  - The server's disk

- What happens when those don't match?

Concurrency Control

- Types of concurrency in SQL Server
  - Read Only
    - Updates are not allowed. Use the last known good data
  - Optimistic Read/Write
    - Assumes row contention is unlikely.
  - Pessimistic Read/Write
    - Assumes row contention is likely and locks the row.

Concurrency Control

```
1  SELECT
2      MemberFirstName
3      , MemberLastName
4      , MemberEmail
5  FROM Members
```

100 %

Results | Messages

	MemberFirstNa...	MemberLastNa...	MemberEmail
1	Matt	Sorum	Matt.Sorum@lookout.net
2	Tim	Alexander	Tim.Alexander@geemail.com
3	Charlie	Benante	Charlie.Benante@yaywho.com
4	Clive	Burr	Clive.Burr@lookout.net
5	Nikko	McBrain	Nikko.McBrain@yaywho.com
6	Adrienne	Davies	Adrienne.Davies@geemail.com
7	Ginger	Fish	Ginger.Fish@yaywho.com
8	Morgan	Lander	Morgan.Lander@geemail.com
9	Mercedes	Lander	Mercedes.Lander@lookout.net
10	Tara	McLeod	Tara.McLeod@yaywho.com
11	Trish	Doan	Trish.Doan@yaywho.com

```
UPDATE Members
SET MemberEmail = 'Trisha.Doan@yaywho.com'
WHERE MemberEmail = 'Trish.Doan@yaywho.com'
```


SYRACUSE UNIVERSITY
SUOS
CULTORES
SCIENTIA
CORONAT