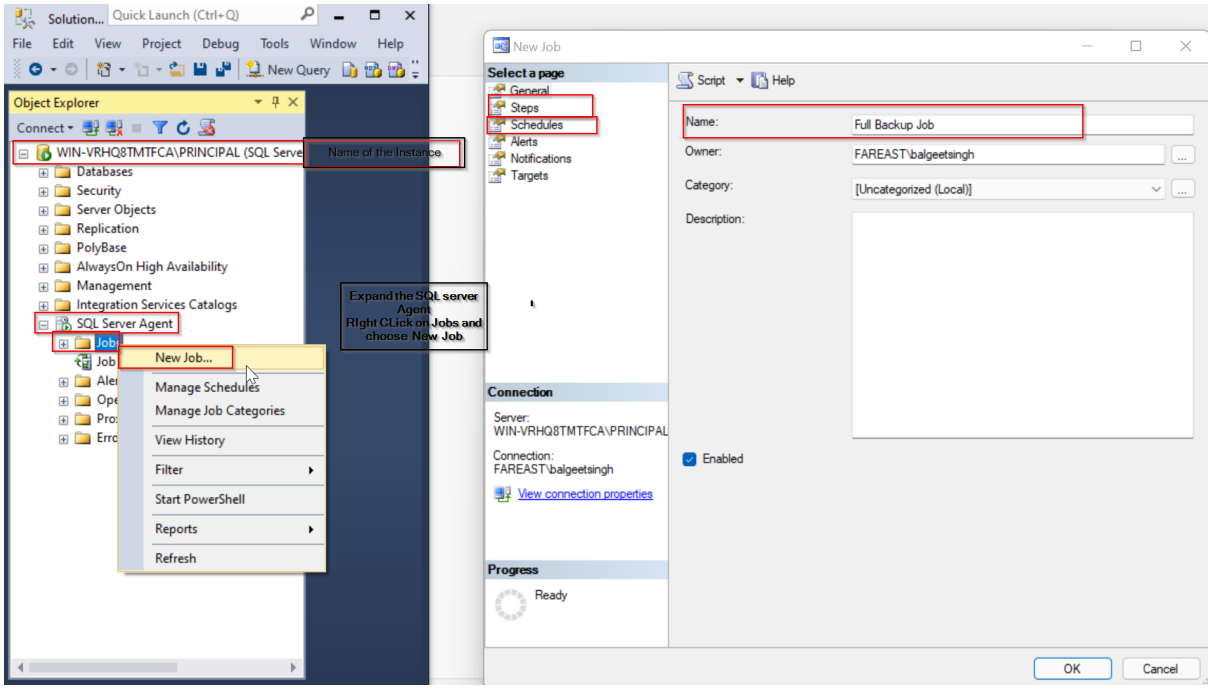
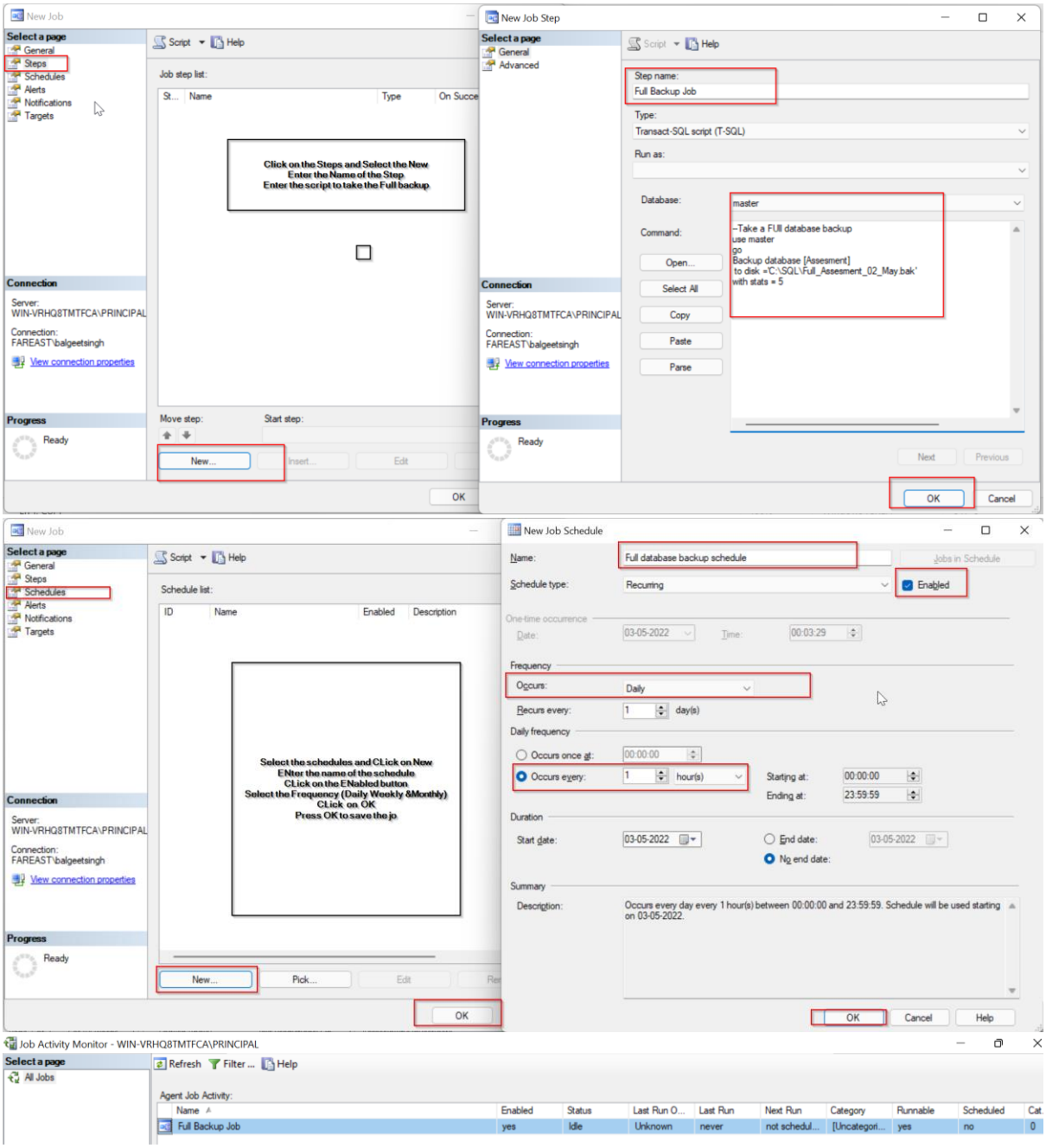
Question no 5→Create a SQL server agent job to take Full, Differential & log Backup

Follow the instruction as below shown in the screenshot to create the multiple Jobs for the Backup.

```
--Take a Full database backup
use master

go

Backup database [Assesment] to disk ='C:\SQL\Full_Assesment_02_May.bak' with stats = 5

--Take a Differential database backup
use master

go

Backup database [Assesment] to disk ='C:\SQL\Diff_Assesment_02_May.bak' with
differential, stats = 5

--Take a Log database backup
--Please note Recovery model of the database should be full
use master

go

Backup log [Assesment] to disk ='C:\SQL\log_Assesment_02_May.trn' with stats = 5
```

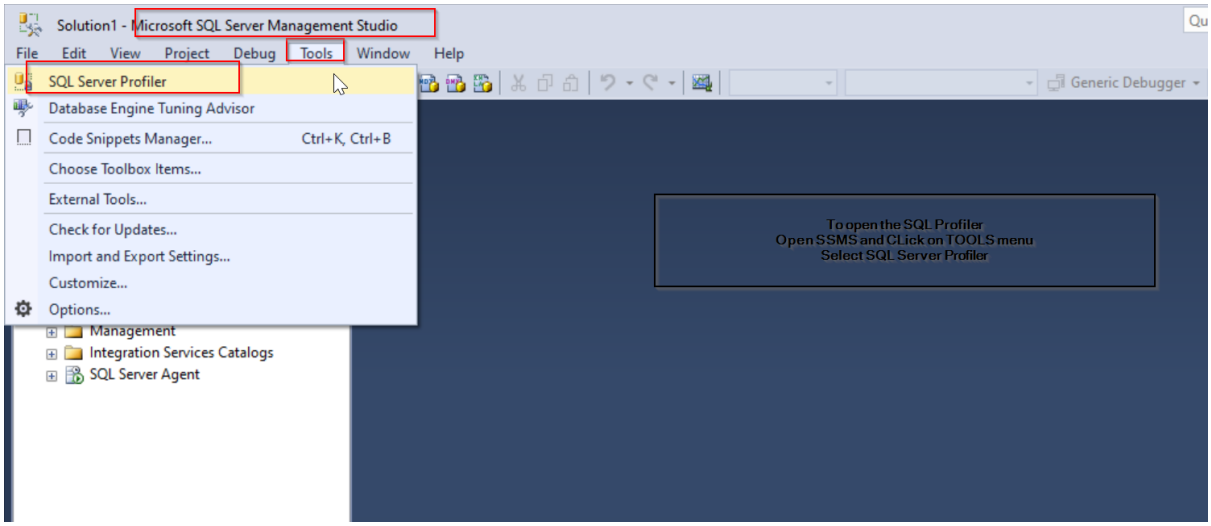


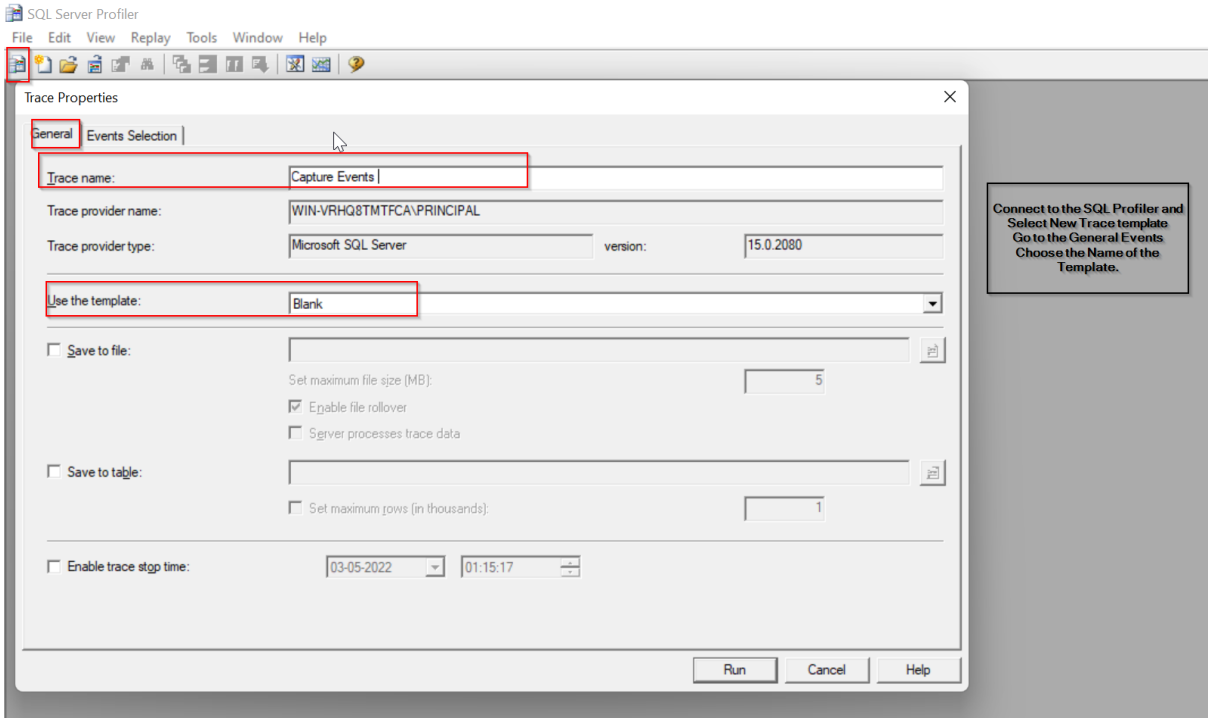

New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Job step list:
St...
Name
Type
On Succe
Click on the Steps and Select the New
Enter the Name of the Step
Enter the script to take the Full backup
Connection
Server:
WIN-VRHQ8TMTFCA\PRINCIPAL
Connection:
FAREAST\balgeetsingh
View connection properties
Progress
Ready
Move step:
Start step:
New...
Insert...
Edit
OK
New Job Step
Select a page
General
Advanced
Script
Help
Step name:
Full Backup Job
Type:
Transact-SQL script (T-SQL)
Run as:
Database:
master
Command:
--Take a FUll database backup
use master
go
Backup database [Asesment]
to disk ='C:\SQL\Full_Asesment_02_May.bak'
with stats = 5
Open...
Select All
Copy
Paste
Parse
Connection
Server:
WIN-VRHQ8TMTFCA\PRINCIPAL
Connection:
FAREAST\balgeetsingh
View connection properties
Progress
Ready
Next
Previous
OK
Cancel
New Job
Select a page
General
Steps
Schedules
Alerts
Notifications
Targets
Script
Help
Schedule list:
ID
Name
Enabled
Description
Select the schedules and CLick on New
ENter the name of the schedule
CLick on the ENabled button
Select the Frequency (Daily Weekly &Monthly)
CLick on OK
Press OK to save the jo
Connection
Server:
WIN-VRHQ8TMTFCA\PRINCIPAL
Connection:
FAREAST\balgeetsingh
View connection properties
Progress
Ready
New...
Pick...
Edit
OK
New Job Schedule
Name:
Full database backup schedule
Jobs in Schedule
Schedule type:
Recurring
Enabled
One-time occurrence
Date:
03-05-2022
Time:
00:03:29
Frequency
Occurs:
Daily
Recurs every:
1
day(s)
Daily frequency
Occurs once at:
00:00:00
Occurs every:
1
hour(s)
Starting at:
00:00:00
Ending at:
23:59:59
Duration
Start date:
03-05-2022
End date:
03-05-2022
No end date:
Summary
Description:
Occurs every day every 1 hour(s) between 00:00:00 and 23:59:59. Schedule will be used starting on 03-05-2022.
OK
Cancel
Help
Job Activity Monitor - WIN-VRHQ8TMTFCA\PRINCIPAL
Select a page
All Jobs
Refresh
Filter ...
Help
Agent Job Activity:
Name
Enabled
Status
Last Run O...
Last Run
Next Run
Category
Runnable
Scheduled
Cat.
Full Backup Job
yes
Idle
Unknown
never
not schedul...
[Uncategori...
yes
no
0

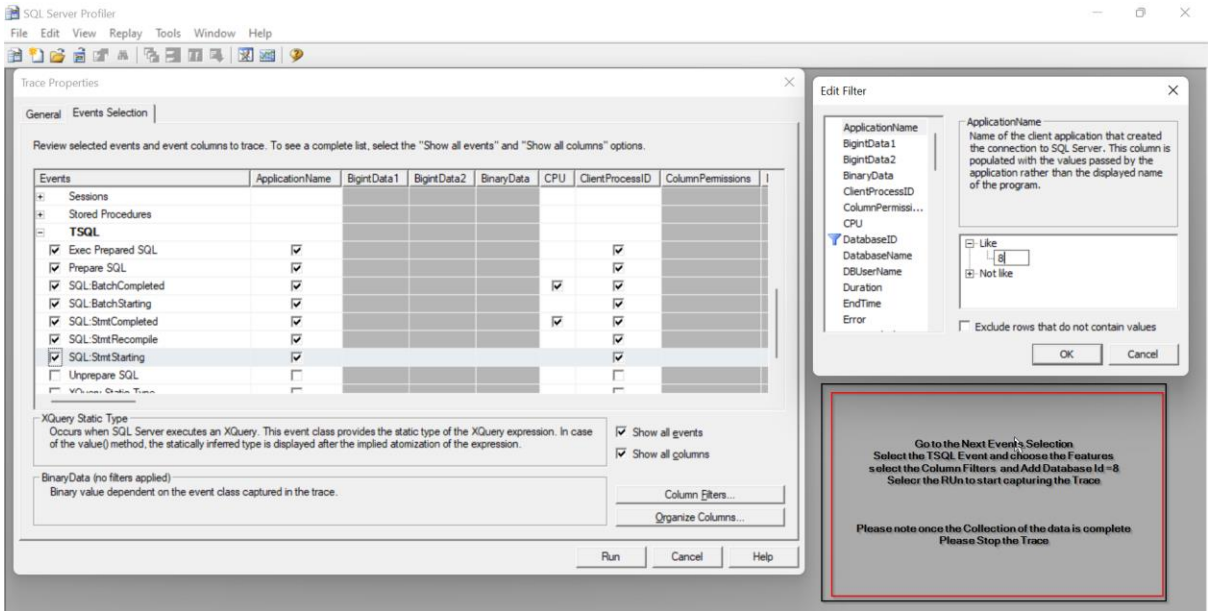
Question 6 → Run profiler to capture the event on the database created

Below are the Stesp to collect the trace

- Open SSMS → Tools→ SQL server Profiler → Connec to the Instance
- Enter the Name of the Trace and choose the Template
- Go to the Next Events Selection
- Select the TSQL Event and choose the Features
- select the Column Filters and Add Database Id =8
- Selecr the RUn to start capturing the Trace
- Please note once the Collection of the data is complete
- Please Stop the Trace



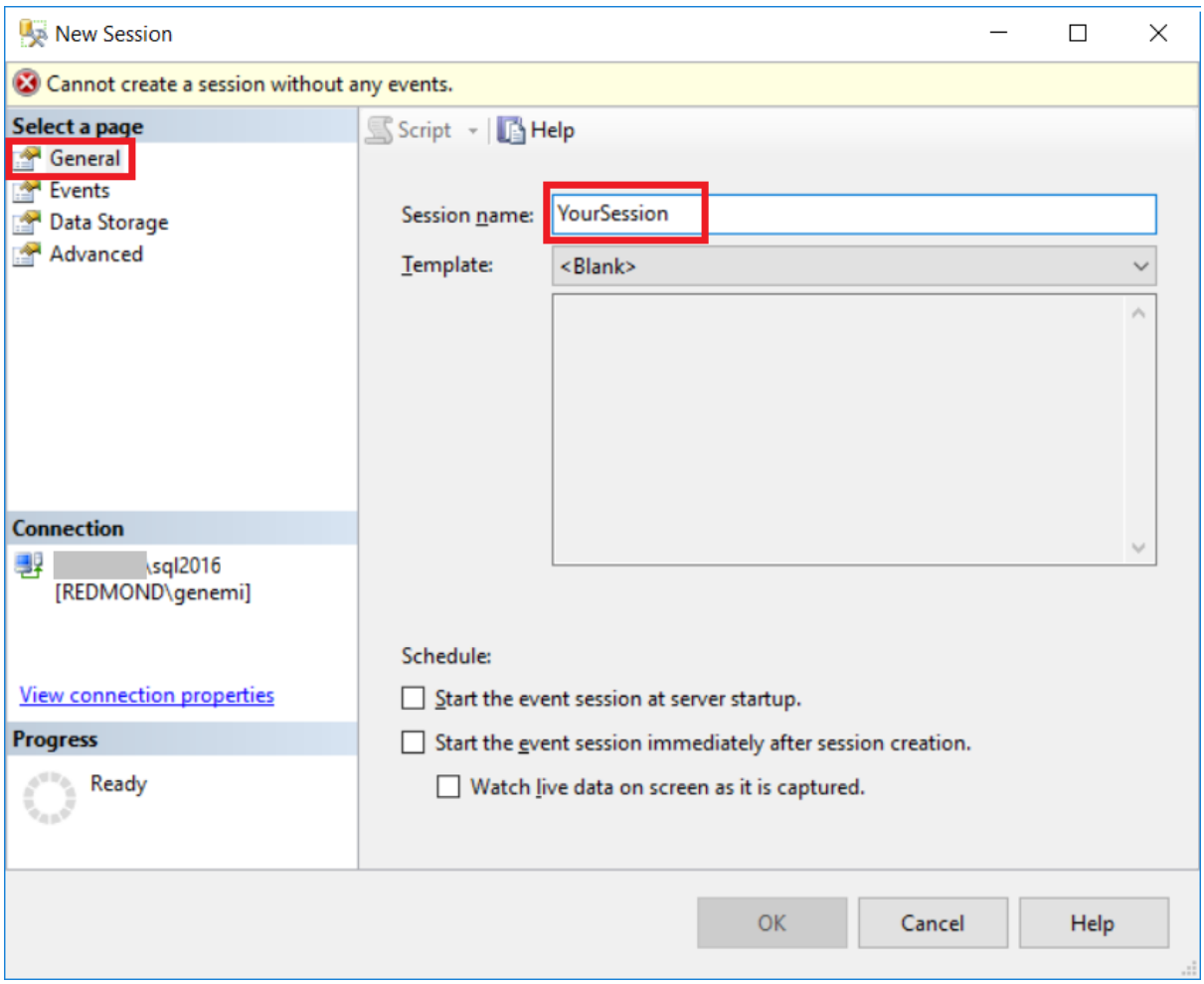




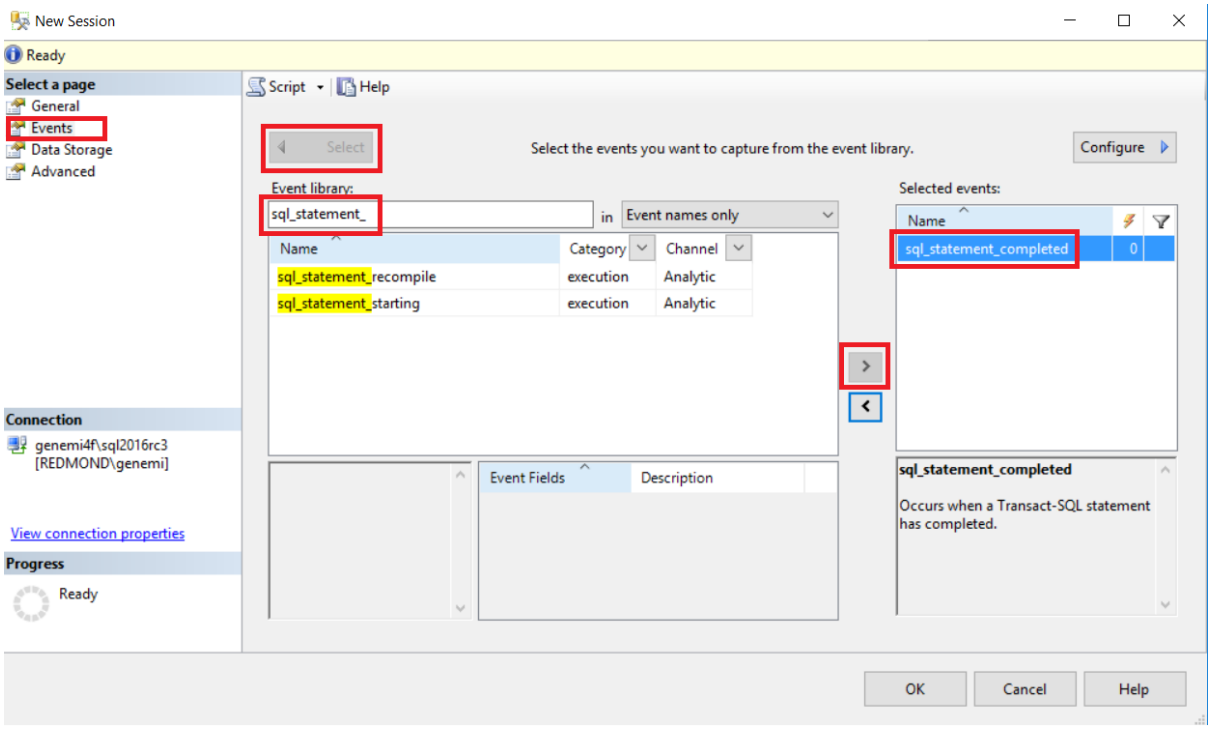
Question 7 TO create the Session to capture the events database attached , database detached

Below are the Step we can use to create the Sessions

1. Connect with SSMS.In the Object Explorer, click **Management** > **Extended Events** > **New Session**. The **New Session** dialog is preferable to the **New Session Wizard**, although the two are similar to each other.
2. In the upper-left, click the **General** page. Then type *YourSession*, or any name you like, into the **Session name** text box. Do *not* press the **OK** button yet, that comes only at the end of the demo.

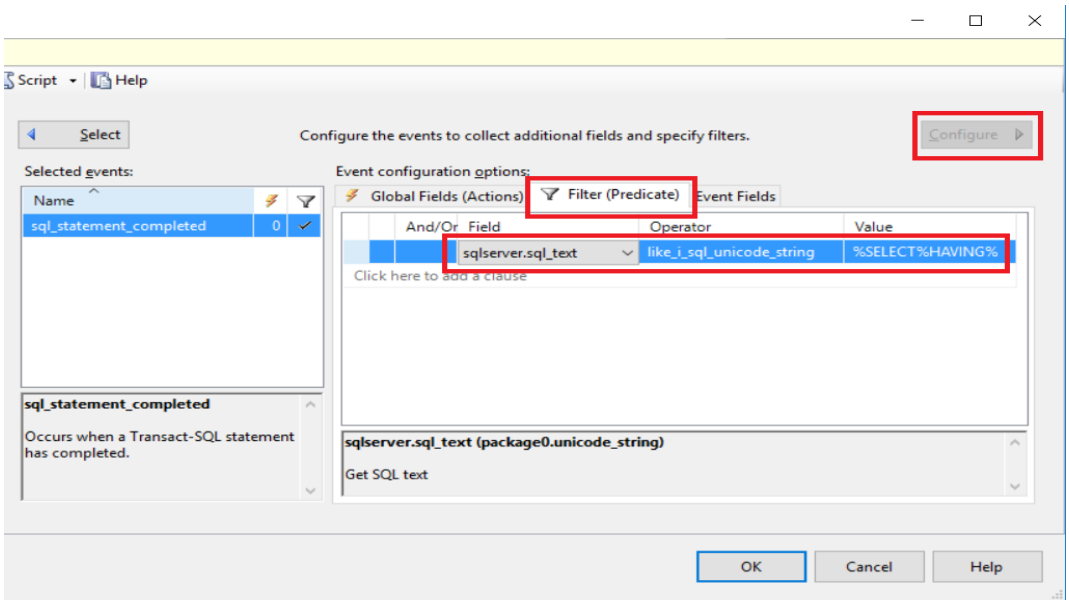


3. In the upper-left, click the **Events** page, and then click the **Select** button.



4. In the **Event library** area, in the drop-down list, choose **Event names only**.
    - Into the text box, type in **sql**, which filters and reduces the long list of available events by using a *contains* operator.
    - Scroll and click the event named **sql_statement_completed**.
    - Click the right arrow button **>** to move the event to the **Selected events** box.
5. Staying on the **Events** page, click the **Configure** button at the far right.

    With the left side chopped off for better display, in the following screenshot you can see the **Event configuration options** area.



6. Click the **Filter (Predicate)** tab. Next, click **Click here to add a clause**, for the intention of capturing all SQL SELECT statements that have a HAVING clause.
7. In the **Field** drop-down list, and choose **sqlserver.sql_text**.
    - For **Operator** choose a LIKE operator.
    - For **Value** type in **%SELECT%HAVING%**.

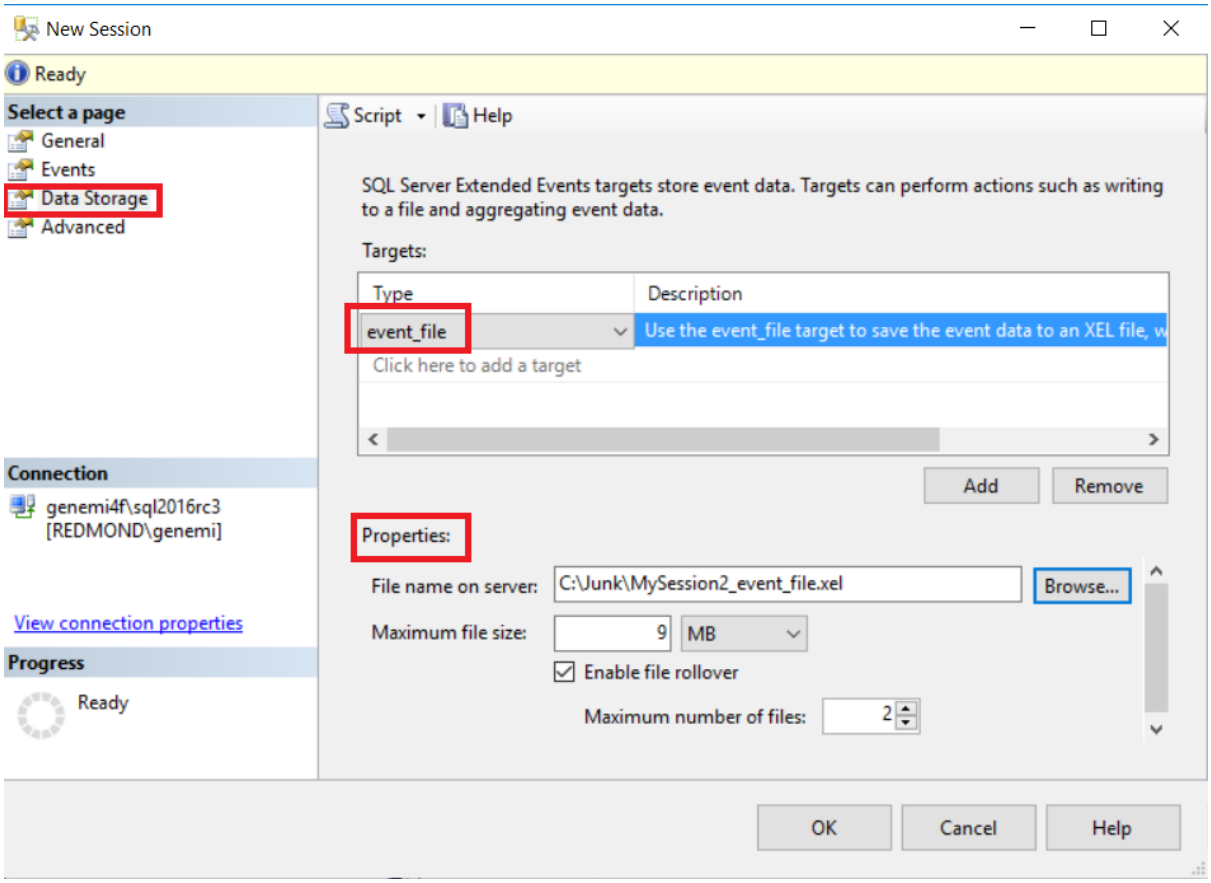
Note

In this two-part name, *sqlserver* is the package name and *sql_text* is the field name. The event we chose earlier, *sql_statement_completed*, must be in the same package as the field we choose.

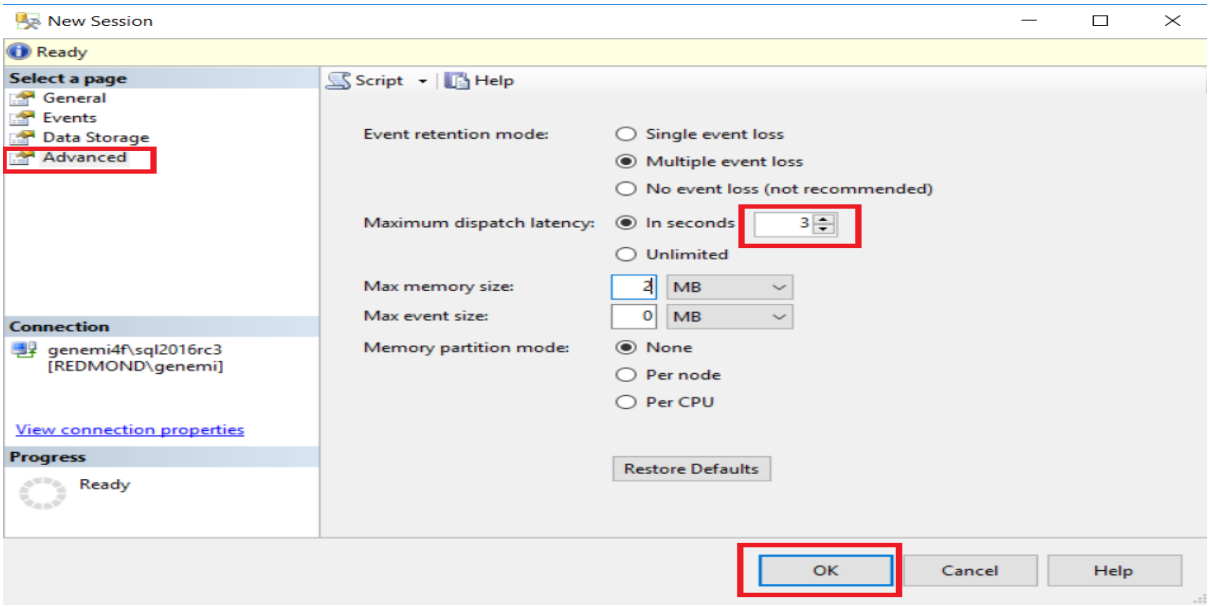
8. In the upper-left, click the **Data Storage** page.
9. In the **Targets** area, click **Click here to add a target**.
    - In the **Type** drop-down list, choose **event_file**.
    - This means the event data will be stored in a file that we can view.

Note

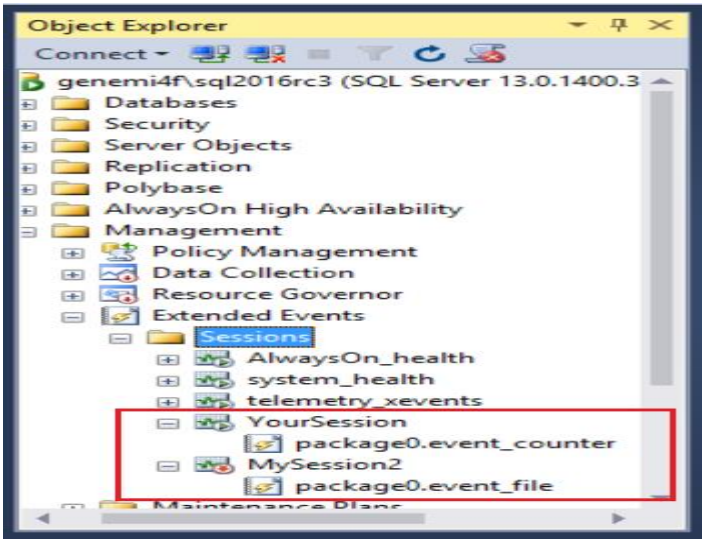
You can't use Azure Blob Storage as a data storage target in an on-premises instance of SQL Server.



10. In the **Properties** area, type in a full path and file name into the **File name on server** text box.
    - The file name extension must be *.xel*.
    - Our little test will need less than 1 MB of file size.



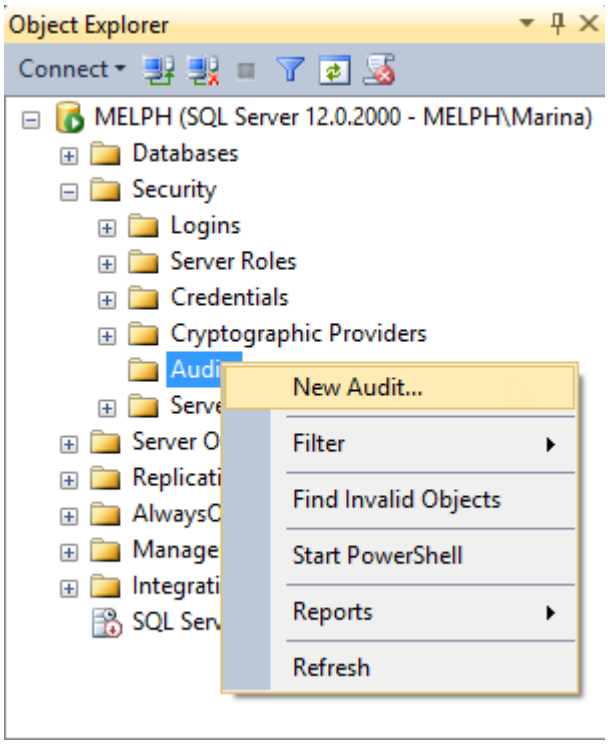
11. In the upper-left, click the **Advanced** page.
    - Reduce the **Maximum dispatch latency** down to 3 seconds.
    - Finally, click the **OK** button at the bottom.
12. Back in the **Object Explorer**, expand **Management** > **Sessions**, and see the new node for **YourSession**.



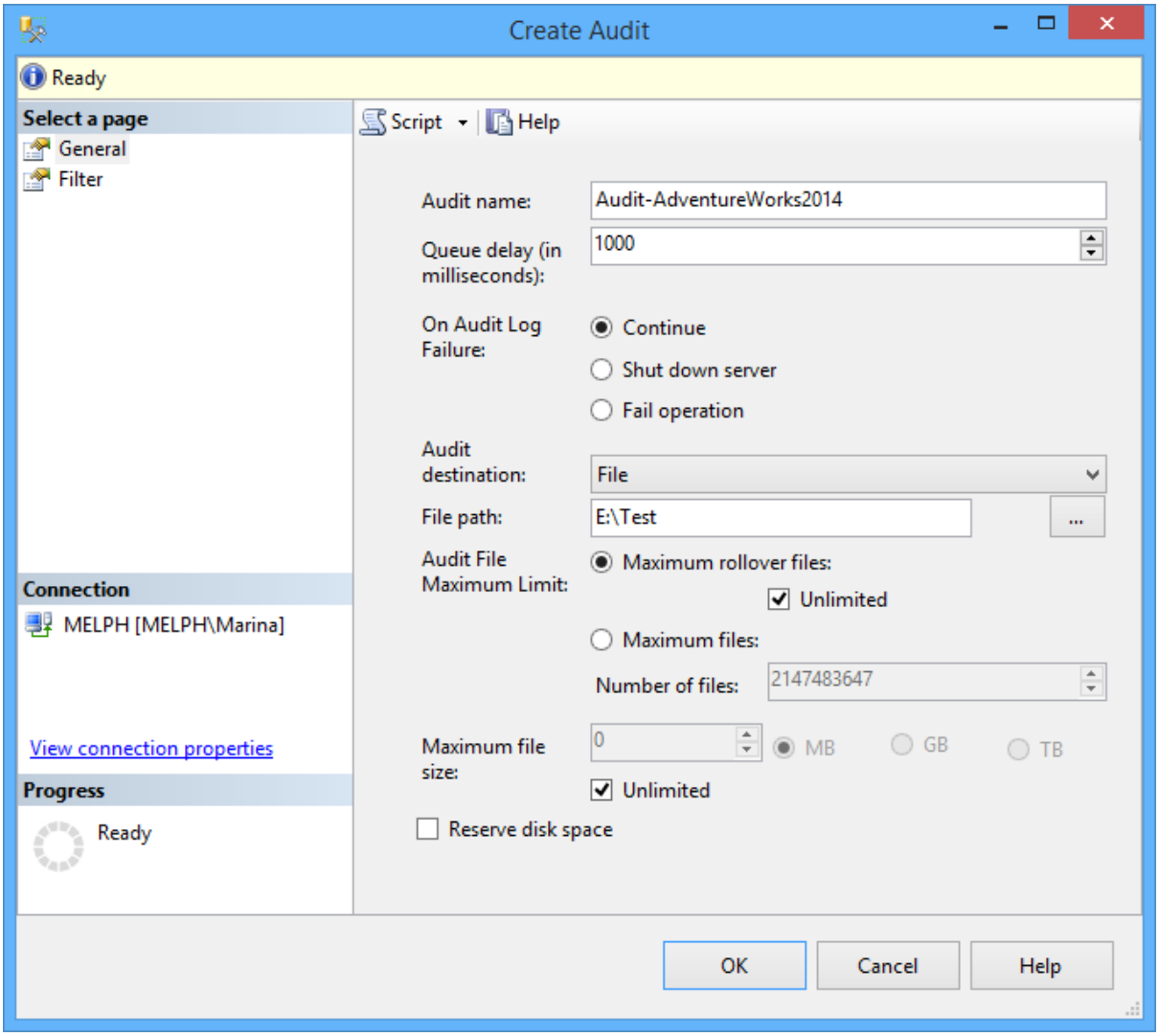
Question -- Audit SQL Server databases

To create a new SQL Server Audit object:

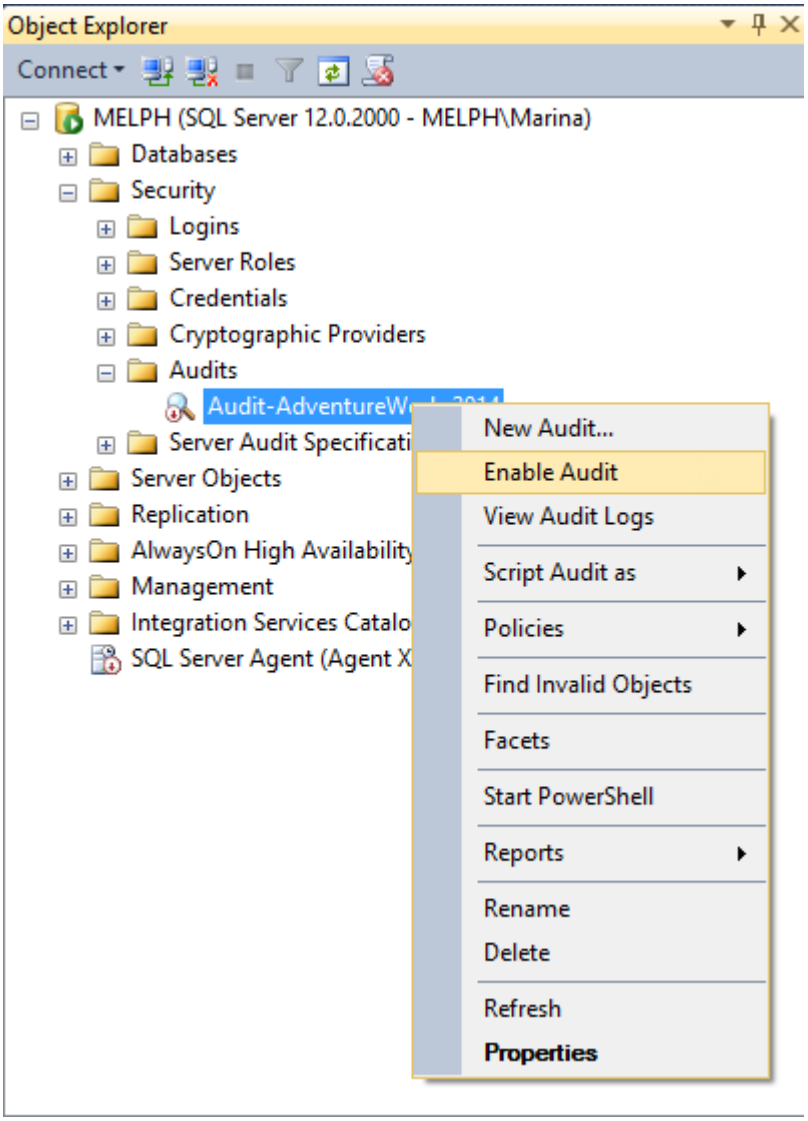
1. Expand Security and right-click Audits in SSMS
2. Select **New Audit**



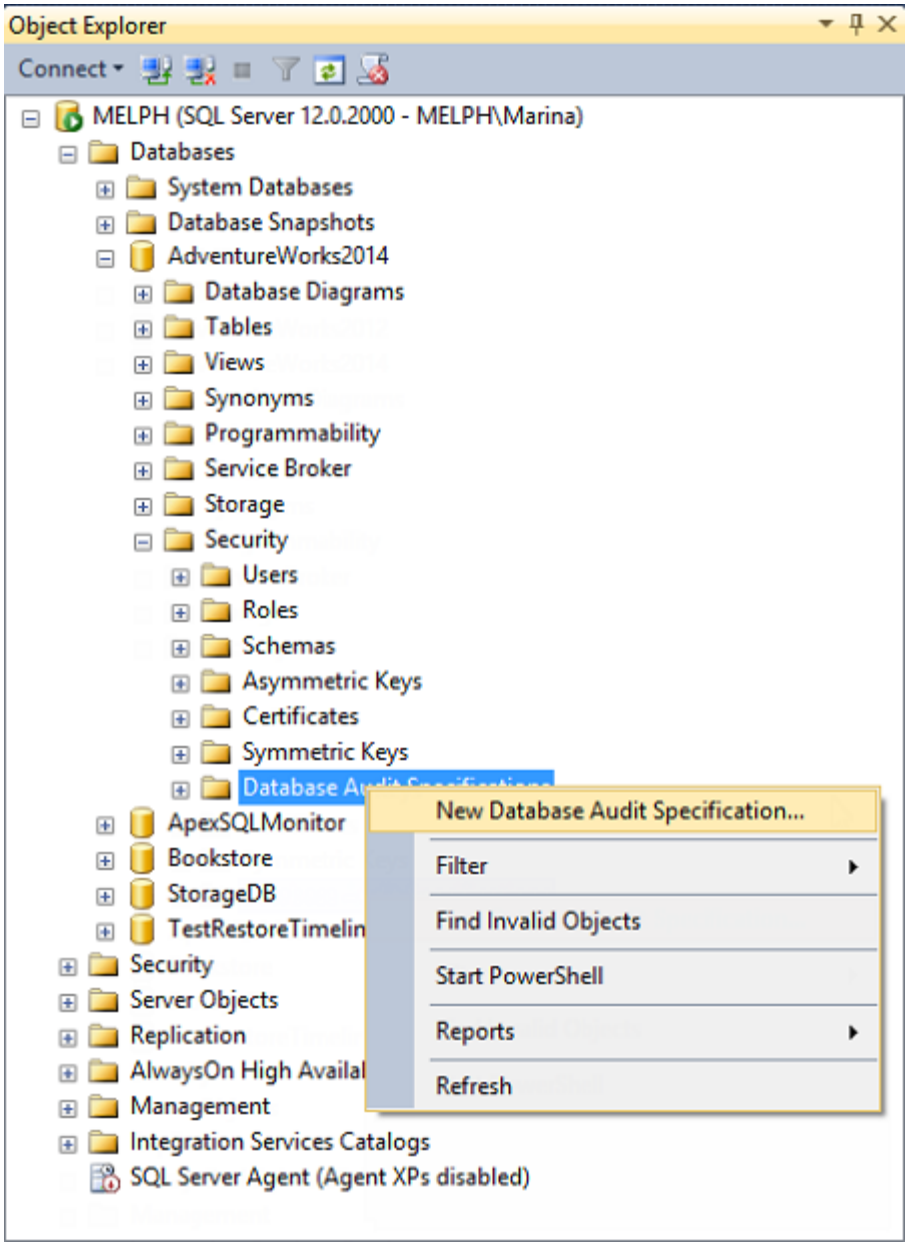
3. You will need to create a name for the audit, and then indicate whether to store the audit data in an application security event log, event log or a file. Finally, indicate a location for the audit file



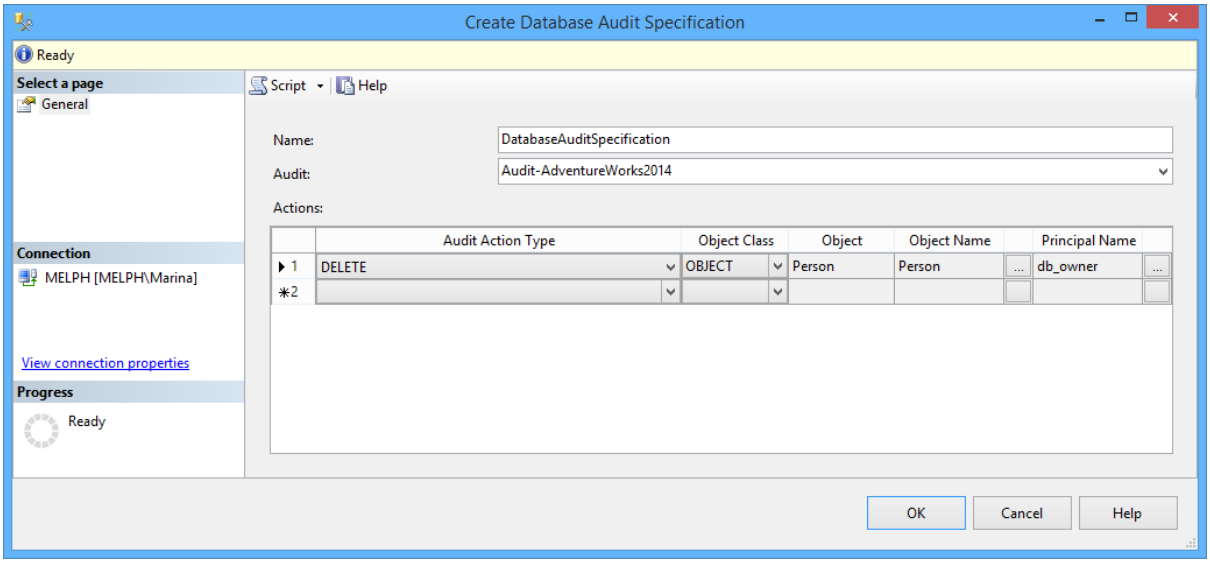
4. Click OK and your audit will appear in the Audits node of the Object Explorer
5. By default, it's disabled. The disabled status is indicated by a red arrow. Right-click and select **Enable Audit**, to enable



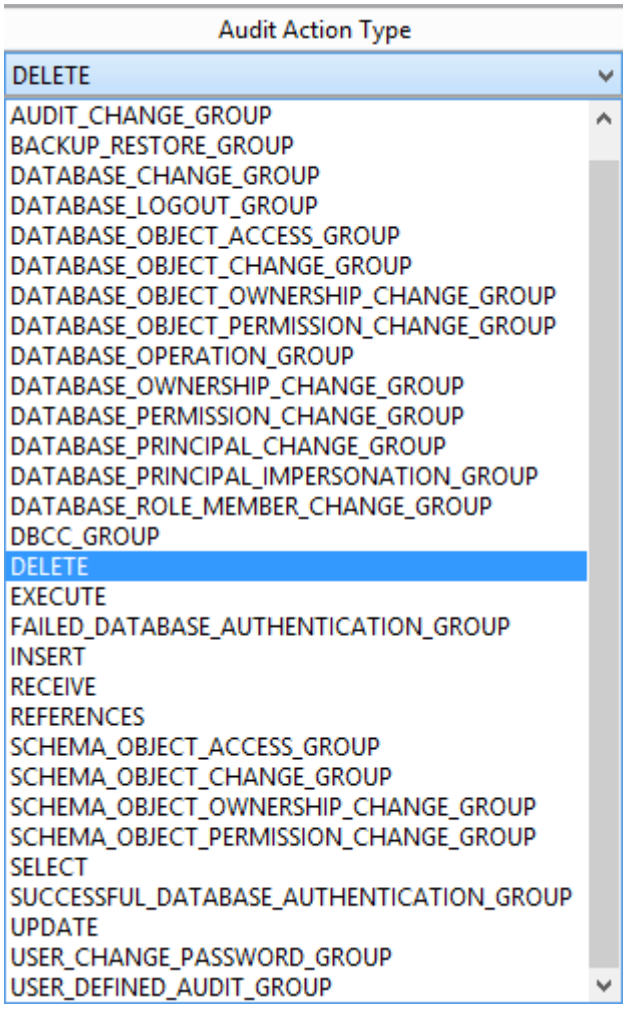
6. Depending on whether you want to audit the activity on an entire SQL Server instance or just a particular database you will choose between Server Audit Specification or Database Audit Specification
7. For a Database Audit Specification, expand the node of the database to audit, go to **Security**, right-click **Database Audit Specifications** and select **New Database Audit**



8. In the Create Database Audit Specification dialog, indicate the specification name, associate the specification with the audit object created in the previous step, specify the activity to audit in the Audit Action Type. For auditing a particular database, indicate the database, object, or schema as an Object Class, the name of the audited object, and the audited login

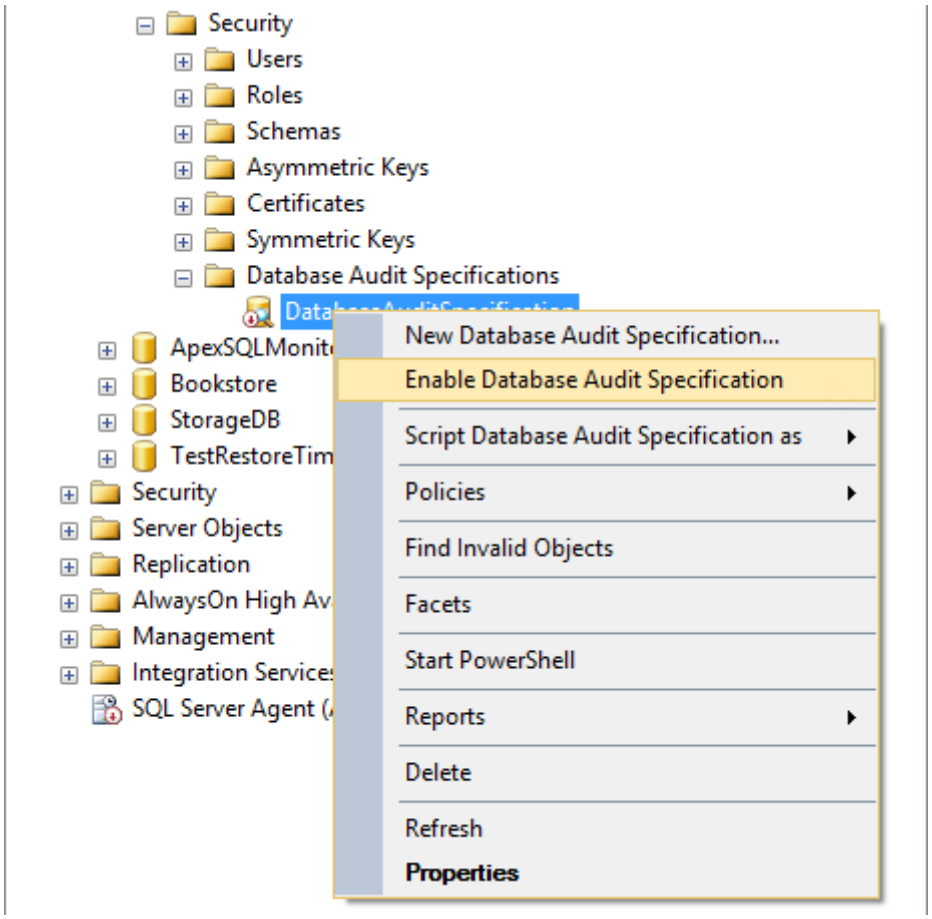


In the drop-down list for Audit Action Type, you can see all actions that can be audited using SQL Server Auditing



In this dialog, you will specify the user accounts to be monitored.

Database audit specifications are disabled, by default. To enable them, select this option in the context menu



Now, all DELETE statements executed against the Person.BusinessEntityAddress will be audited and inserted into files the names of which start with Audit-, such as Audit-AW2012Test_9D93CA4A-8B90-40B8-8B0B-FCBDA77B431D_0_130161593310500000.sqlaudit, and stored in E:\

For high volume databases, it is considered a good practice to save audited info to a file. This file can't be opened directly though, even with a hex editor. To view it, use fn_get_audit_file

For example:

```
SELECT event_time,action_id, statement, database_name, server_principal_name
 FROM fn_get_audit_file( 'E:\Test\Audit-*.sqlaudit' , DEFAULT , DEFAULT);
```

shows the following results:

	event_time	action_id	statement	database_name	server_principal_name
1	2013-06-19 23:48:51.0799953	AUSC			Fujitsu\Milena

The results will include columns but not the actual deleted data. The user, who made the deletion, and when the deletion was made are the main audit elements here, which is a disadvantage for users who need more comprehensive audit data.