CEPSUNI

www.ceps.uni.edu.pe

UNIDAD 04 – Separata 03

AUDITORIA EN SQL SERVER

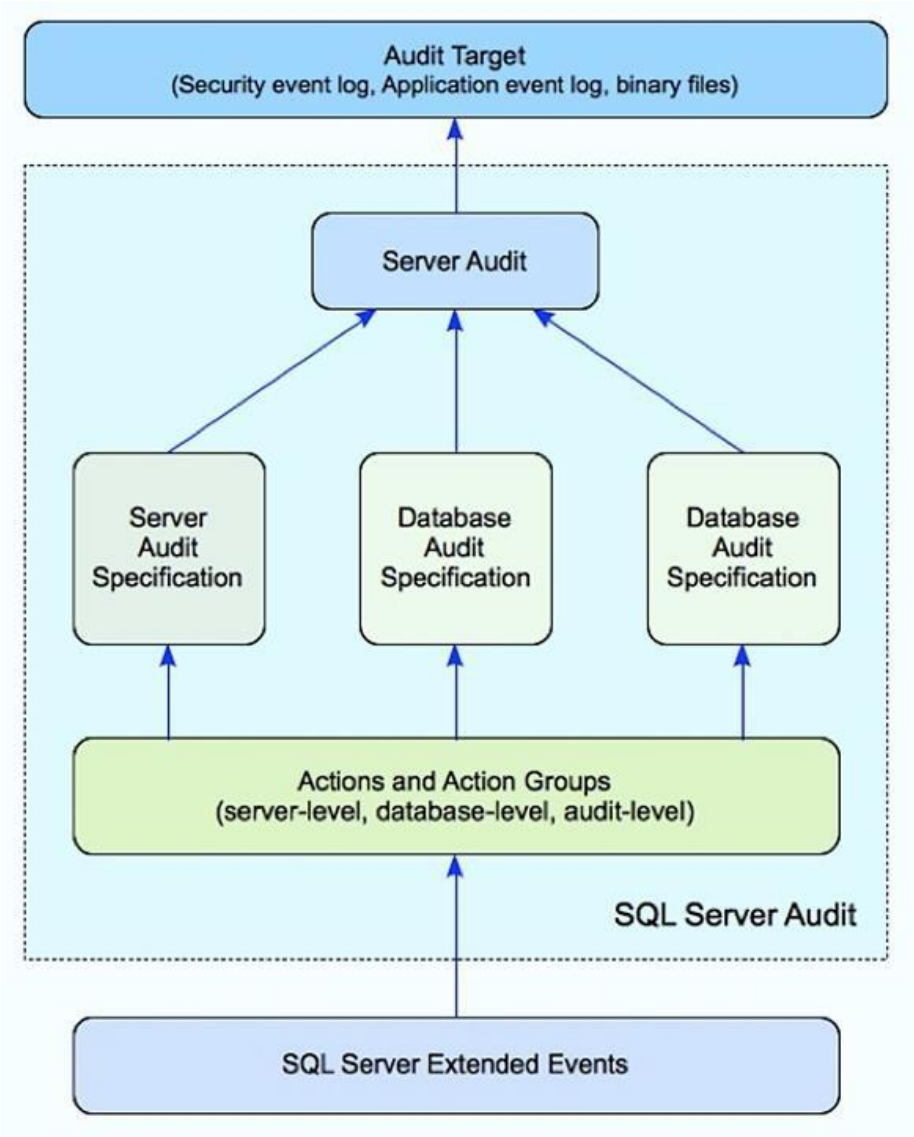
GUSTAVO CORONEL

www.youtube.com/c/DesarrollaSoftware

gcoronel@uni.edu.pe

Temas

1 INTRODUCCIÓN

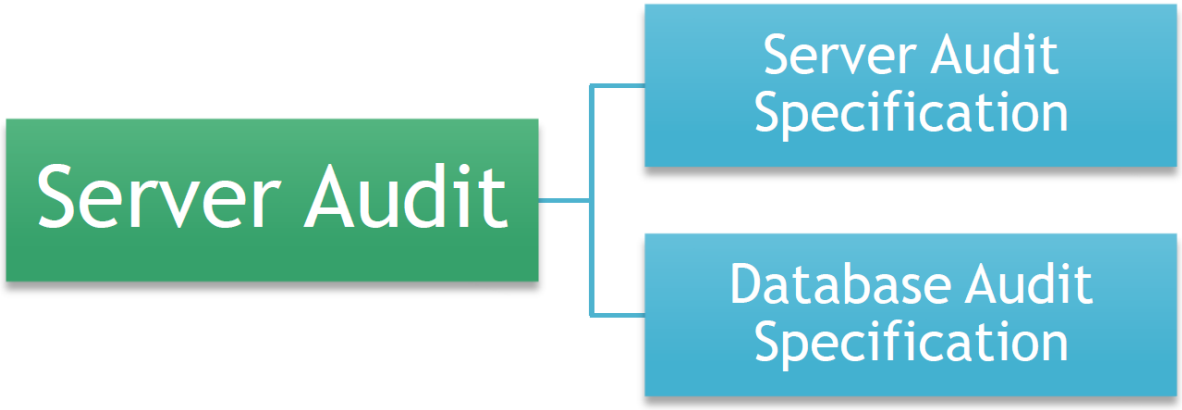


En el área de seguridad informática en muchos casos es necesario poder contar con auditorias de las operaciones que se ejecutan en el motor de base de datos.

Es importante saber que para usar todas las funcionalidades de auditoria se requiere edición Enterprise de SQL Server ya que la Standard tiene la funcionalidad, pero de forma limitada.

En esta guía se explica paso a paso como se implementan las auditorias nativas que tiene SQL Server, pues consultar información oficial en: https://docs.microsoft.com/en-us/sql/relational-databases/security/auditing/sql-server-audit-database-engine.

2 COMPONENTES



Las auditorias contienen los siguientes componentes:

Server Audit	Es el objeto principal, aquí se definen por ejemplo los lugares de persistencia de las auditorias (file, security Log o Application Log. Se pueden crear más de uno a nivel instancia
Server Audit Specifications	Permite auditar eventos a nivel instancia por ej. (Login Fail, créate database, etc.) Es necesario que exista un Server Audit y se pueden crear más de uno
Database Audit Specifications	Permite auditar eventos a nivel base de datos por ej. (Select, insert, alter, etc) Es necesario que exista un Server Audit y se crean a nivel base de datos, por cada una de las que se desea tener este tipo de auditoria

3 PASO 1: SERVER AUDIT

El primer paso es crear los **SERVER AUDIT**, en ellos se podrá definir la persistencia y otras configuraciones adicionales. En esta guía crearemos dos Server Audit, uno para los eventos de instancia y otro para los de base de datos.

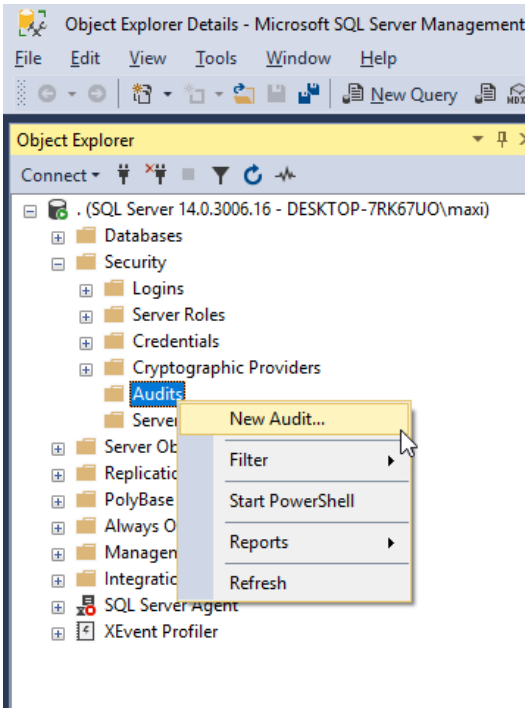
3.1 PERMISOS DE SEGURIDAD

Es necesario que tomes en cuenta lo siguiente:

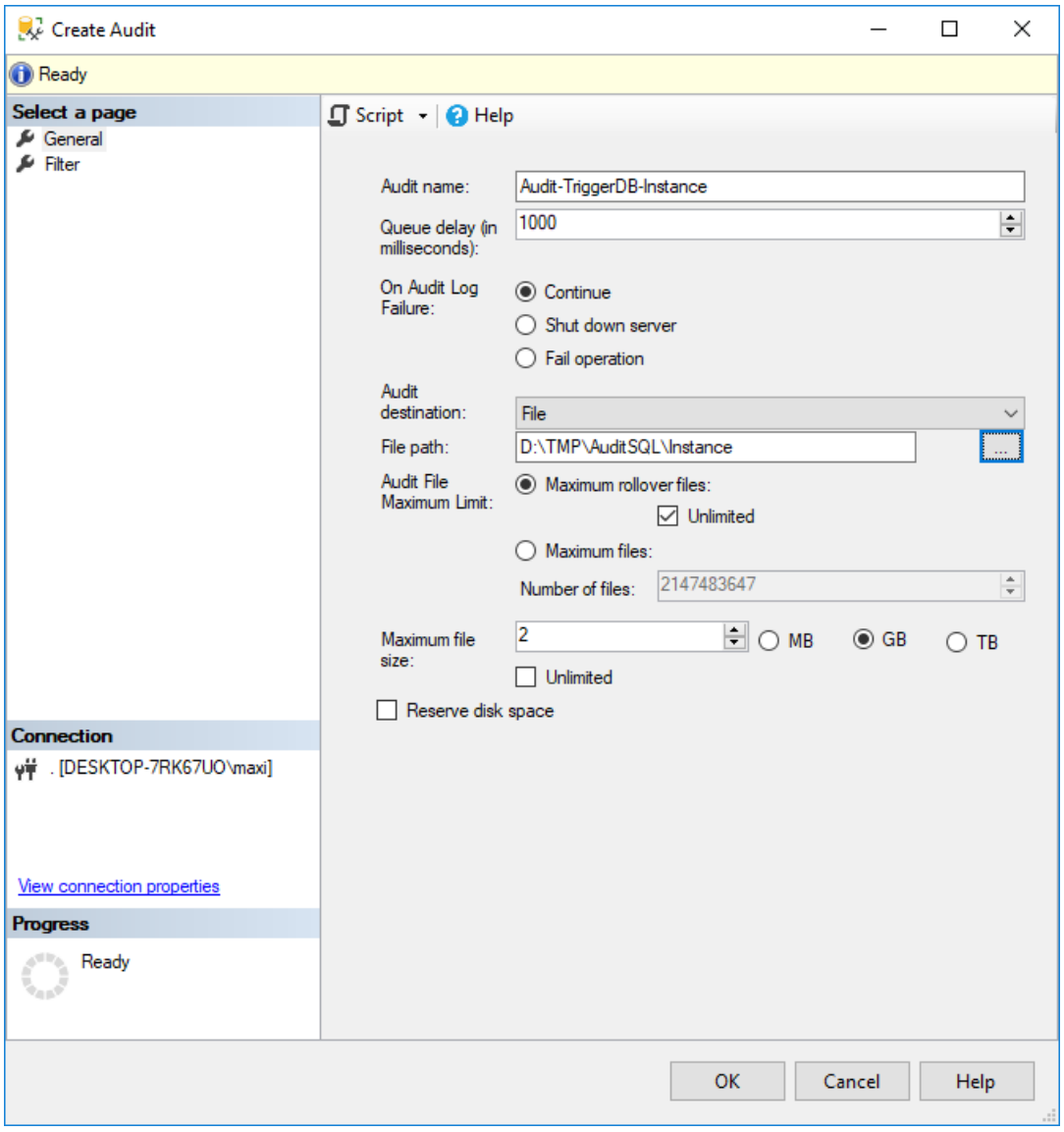
- Para crear, modificar o quitar una auditoría de servidor, las entidades de seguridad deben tener el permiso **ALTER ANY SERVER AUDIT** o **CONTROL SERVER**.
- Los usuarios con el permiso **ALTER ANY SERVER AUDIT** pueden crear especificaciones de auditoría de servidor y enlazarlas a cualquier auditoría.
- Una vez creada una especificación de auditoría de servidor, las entidades de seguridad que cuenten con los permisos **CONTROL SERVER** o **ALTER ANY SERVER AUDIT**, así como la cuenta **sysadmin**, o las entidades de seguridad que tengan acceso explícito a la auditoría podrán ver dicha especificación.

3.2 CREAR SERVER AUDIT PARA EVENTOS DE INSTANCIA

En el SSMS iniciar la creación del Server Audit como se ilustra a continuación:



Luego completas los datos del Server Audit, como se ilustra a continuación:

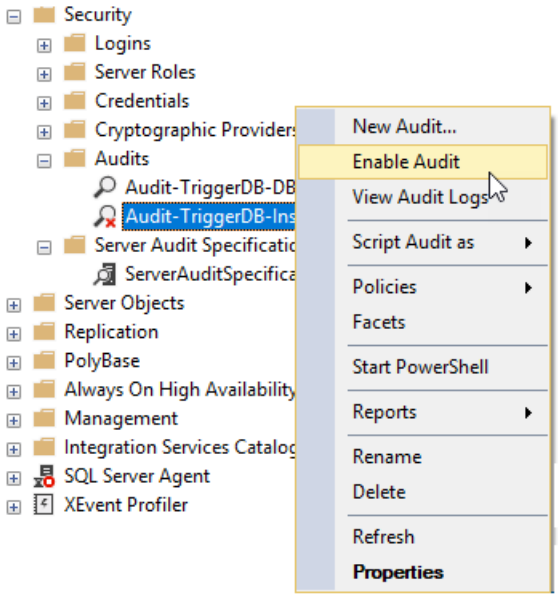


A continuación, tienes información sobre los parámetros del Server Audit:

Audit Name	Aquí debemos escribir el nombre del Server Audit, en nuestro caso Audit-TriggerDB-Instance ya que aquí solo la usaremos para eventos de instancia y no base de datos. Si bien se puede crear un solo archivo para todo, en una buena práctica y más ordenado tener dos o más Server Audit, separando los de instancia de los de base de datos
Queue delay	Especifica la cantidad de tiempo, en milisegundos, que puede transcurrir antes de exigir que se procesen las acciones de auditoría. El valor 0 indica la entrega sincrónica. El valor mínimo predeterminado es 1000 (1 segundo). El máximo es 2.147.483.647 (2.147.483,647 segundos, o 24 días, 20 horas, 31 minutos y 23,647 segundos).

On Audit Log Failure	**Continue:** SQL Server Las operaciones de continúan. Los registros de auditoría no se conservan. La auditoría continúa intentando el registro de eventos y se reanudará si se resuelve la condición de error. La selección de la opción Continuar puede permitir que una actividad no se audite, con lo que se infringirían las directivas de seguridad. Seleccione esta opción cuando la operación de continuación del Motor de base de datos sea más importante que el mantenimiento de una auditoría completa. Esta es la selección predeterminada **Shut Down Server:** Fuerza el apagado del servidor cuando la instancia de servidor que escribe en el destino no puede escribir datos en el destino de la auditoría. Para poder usarlo, es preciso utilizar un inicio de sesión con el permiso SHUTDOWN . Si el inicio de sesión no tiene dicho permiso, la función generará un error y se mostrará un mensaje de error. No se producirán eventos auditados. Seleccione esta opción si un error de auditoría puede poner en peligro la seguridad o la integridad del sistema. **Fail Operation:** En los casos en que SQL Server Audit no puede escribir en el registro de auditoría, esta opción haría que las acciones de base de datos produjesen un error si generasen eventos auditados. No se producirán eventos auditados. Las acciones que no producen eventos auditados pueden continuar. La auditoría continúa intentando el registro de eventos y se reanudará si se resuelve la condición de error. Seleccione esta opción si el mantenimiento de una auditoría completa es más importante que el acceso total al Motor de base de datos.
Audit Destination	**File:** El destino serán archivos binarios **Security Log:** Los eventos se escriben en el Security Log de Windows **Application Log:** Los eventos se escriben en el Application Log de Windows **Nota:** para todos los casos la cuenta de servicio del motor de base de datos debe tener los permisos adecuados ya sea para escribir en las carpetas o en los eventos del SO.
File path	La ruta donde se guardarán los archivos, para el primer ejemplo hemos seleccionado **"D:\TMP\AuditSQL\Instance"** ya que ahí guardaremos los archivos para los eventos de instancia.
Maximum file Size	Por defecto esta opción deja tener tamaño ilimitado, en nuestro caso y en base a las buenas practicas configuraremos que los archivos no puedan tener más de 2GB cada uno.

Es necesario habilitar los Server Audit, tal como se ilustra a continuación:

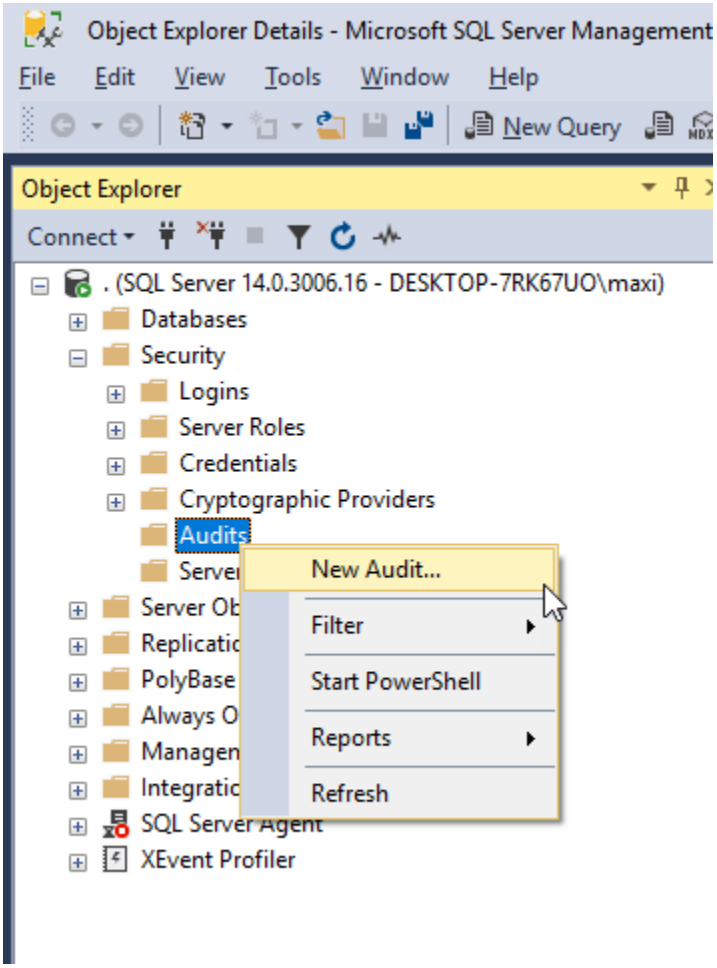


El siguiente código TSQL es la representación de lo que hemos hecho anteriormente

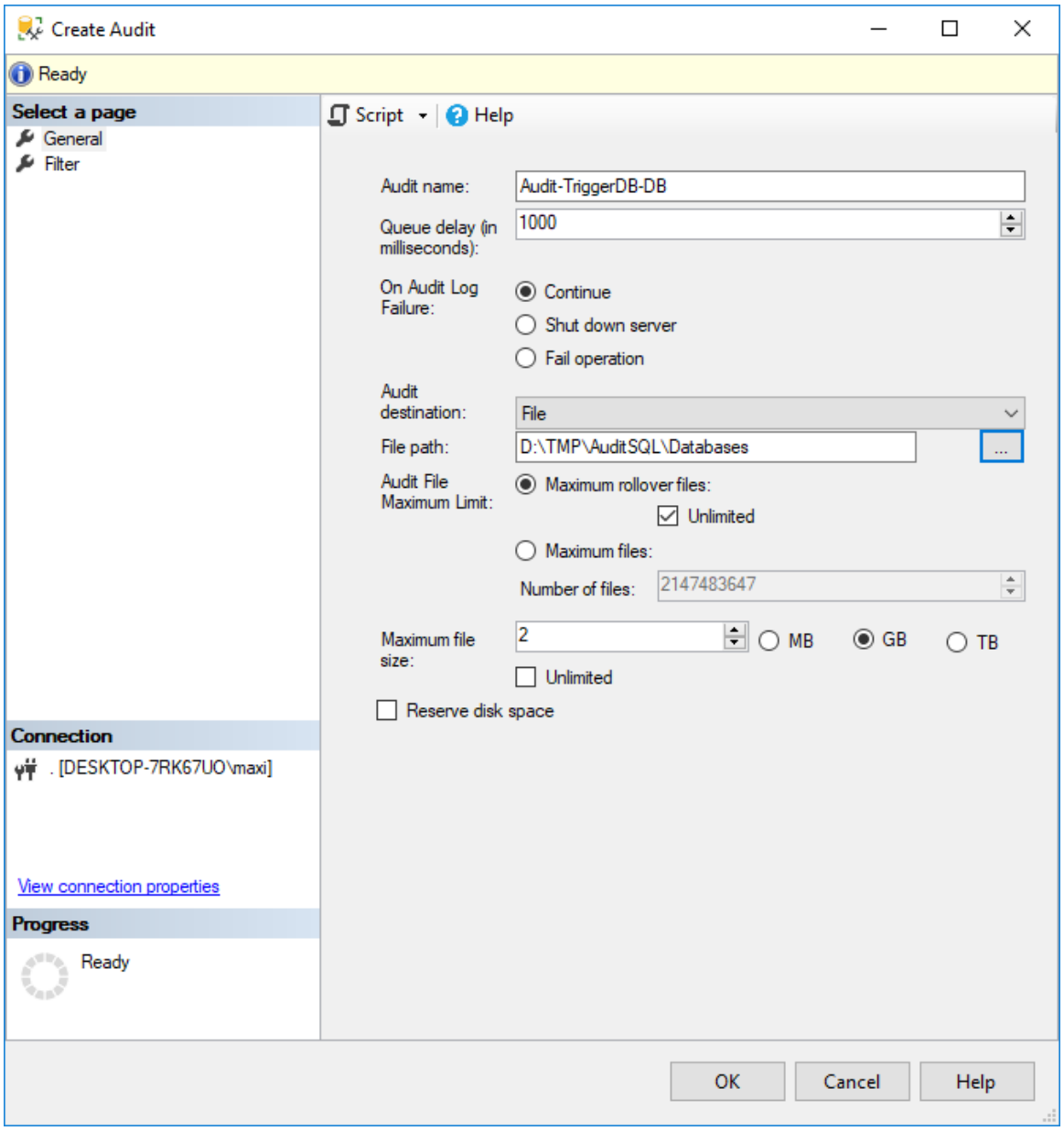
```
USE [master]
GO

CREATE SERVER AUDIT [Audit-TriggerDB-Instance]
TO FILE
(
   FILEPATH = N'D:\TMP\AuditSQL\Instance'
   MAXSIZE = 2 GB
   MAX_ROLLOVER_FILES = 2147483647
   RESERVE_DISK_SPACE = OFF
)
WITH
(
   QUEUE_DELAY = 1000
   ON_FAILURE = CONTINUE
)
GO

ALTER SERVER AUDIT [Audit-TriggerDB-Instance]
WITH (STATE = ON);
```

3.3 CREAR SERVER AUDIT PARA EVENTOS DE BASE DE DATOS

En el SSMS iniciar la creación del Server Audit como se ilustra a continuación:



Luego se debe especificar los datos del Server Audit:



A continuación, tienes código correspondiente:

```
USE [master]
GO

CREATE SERVER AUDIT [Audit-TriggerDB-DB]
TO FILE
(
   FILEPATH = N'D:\TMP\AuditSQL\Databases'
   MAXSIZE = 2 GB
   MAX_ROLLOVER_FILES = 2147483647
   RESERVE_DISK_SPACE = OFF
)
WITH
(
   QUEUE_DELAY = 1000
   ON_FAILURE = CONTINUE
)
GO

ALTER SERVER AUDIT [Audit-TriggerDB-DB]
WITH (STATE = ON);
```

4 PASO 2: SERVER AUDIT SPECIFICATION

En este paso vamos a crear un **Server Audit Specification** para así poder auditar los eventos que nos interesa a nivel instancia.

En el siguiente enlace se encuentran los distintos eventos que se pueden auditar a nivel instancia y asignarlos al **Server Audit Specification**

https://docs.microsoft.com/es-mx/sql/relational-databases/security/auditing/sql-server-audit-action-groups-and-actions

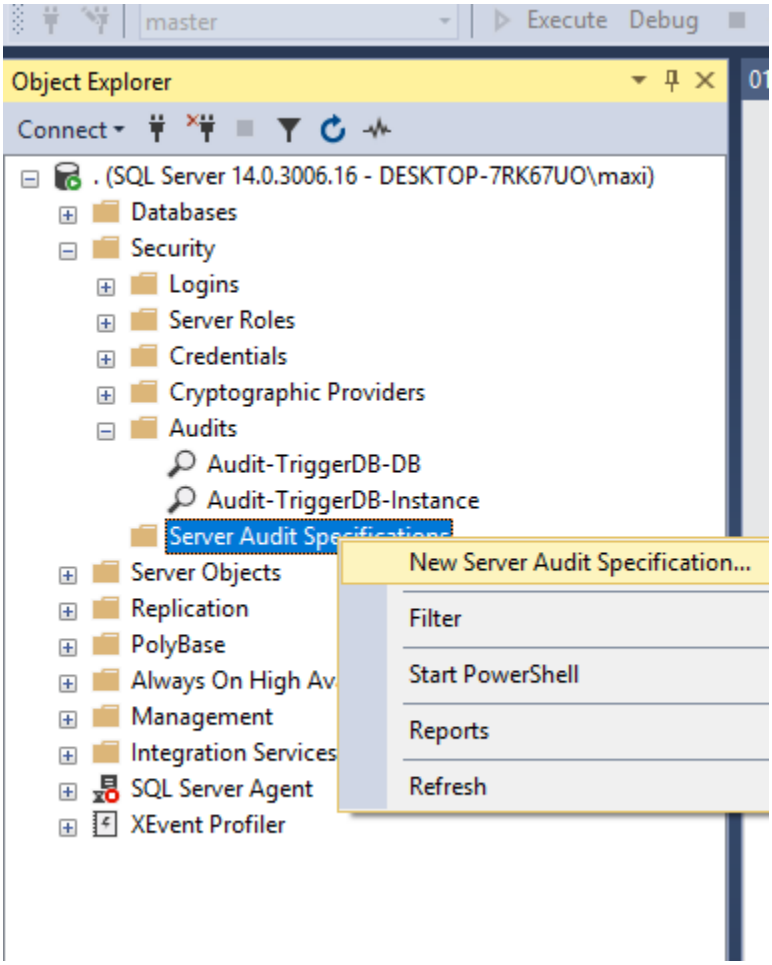
4.1 PERMISOS DE SEGURIDAD

Es necesario tener en cuenta los criterios de seguridad necesarios:

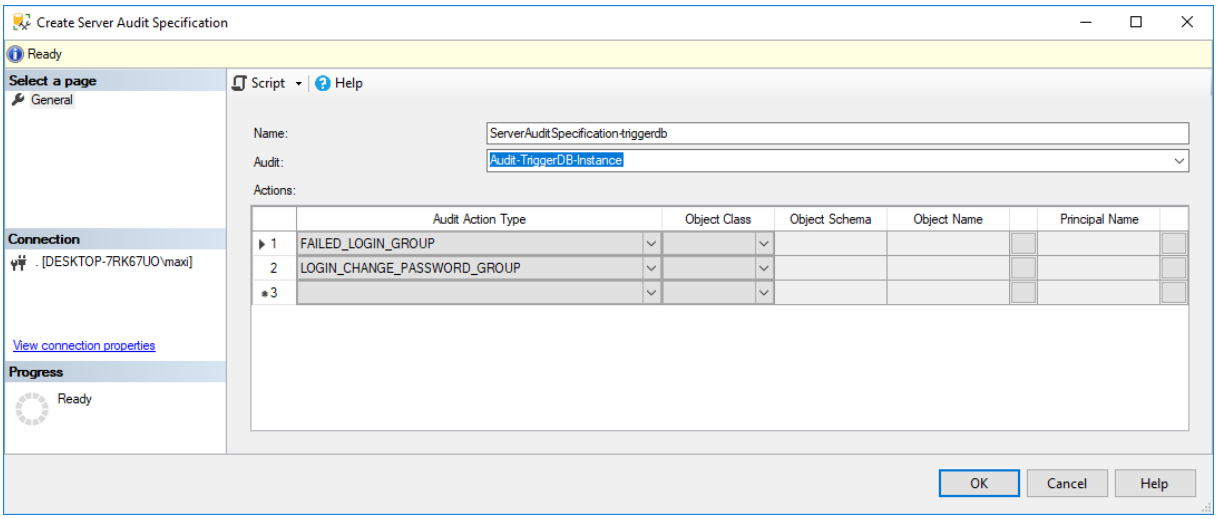
- Para crear, modificar o quitar una auditoría de servidor, las entidades de seguridad deben tener el permiso ALTER ANY SERVER AUDIT o CONTROL SERVER.
- Los usuarios con el permiso ALTER ANY SERVER AUDIT pueden crear especificaciones de auditoría de servidor y enlazarlas a cualquier auditoría.
- Una vez creada una especificación de auditoría de servidor, las entidades de seguridad que cuenten con los permisos CONTROL SERVER o ALTER ANY SERVER AUDIT, así como la cuenta sysadmin, o las entidades de seguridad que tengan acceso explícito a la auditoría podrán ver dicha especificación.

4.2 CREANDO SERVER AUDIT SPECIFICATION

Desde el SSMS inicias la creación del Server Audir Specification:



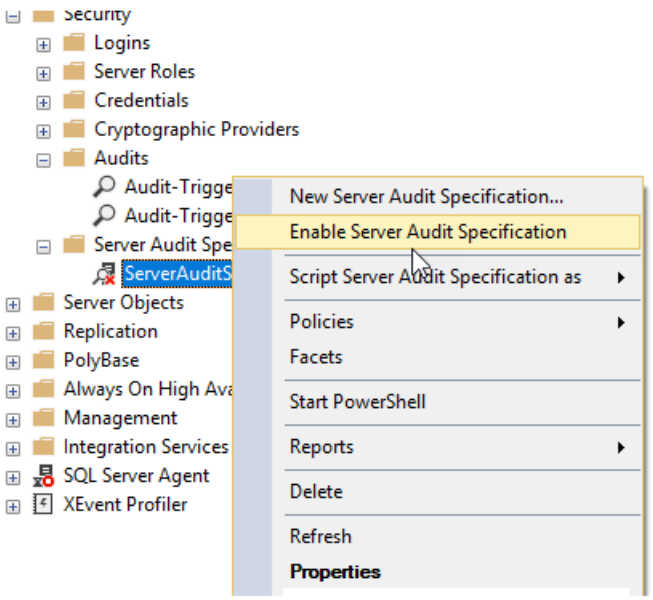
Luego debe completar la información:



La siguiente información te ayudará a ingresar la información requerida:

Name	Aquí ingresaremos el nombre de nuestro Server Audit Specification , en nuestro ejemplo “ServerAuditSpecification-triggerdb”.
Audit	Aquí debemos seleccionar el Server Audit en el cual se persistirán los eventos (los hemos creado en el paso anterior) , en nuestro ejemplo usaremos “Audit-TriggerDB-Instance”.
Actions	Aquí seleccionaremos los eventos a auditar, para este ejemplo solo hemos elegido dos.

También debe habilitarlo, como se ilustra a continuación:



A continuación, tienes el código respectivo:

```
USE [master]
GO

CREATE SERVER AUDIT SPECIFICATION
[ServerAuditSpecification-triggerdb]
FOR SERVER AUDIT [Audit-TriggerDB-Instance]
ADD (FAILED_LOGIN_GROUP),
ADD (LOGIN_CHANGE_PASSWORD_GROUP)
GO

ALTER SERVER AUDIT SPECIFICATION
[ServerAuditSpecification-triggerdb]
WITH (STATE = ON);
GO
```

5 PASO 3: DATABASE AUDIT SPECIFICATION

A diferencia del Server Audit Specification, los Database Audit Specification se deben crear por cada una de las bases de datos que se desee auditar.

Este objeto reside en la metada de la base de datos con lo cual un Backup / Restore también incluirá estas definiciones.

5.1 PERMISOS DE SEGURIDAD

- Los usuarios con el permiso ALTER ANY DATABASE AUDIT pueden crear las especificaciones de auditoría de base de datos y enlazarlas a cualquier auditoría.
- Después de crearse una especificación de auditoría de base de datos, podrá ser vista por las entidades de seguridad que cuenten con los permisos CONTROL SERVER o ALTER ANY DATABASE AUDIT, o por la cuenta sysadmin

5.2 ALGUNAS RECOMENDACIONES

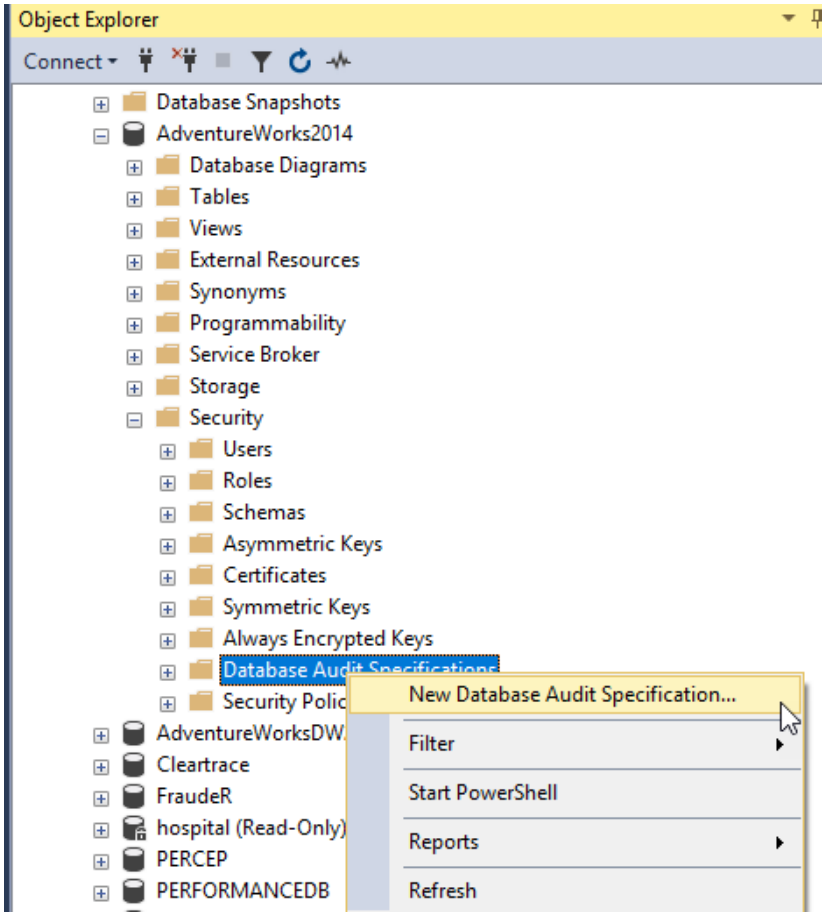
En la mayoría de las empresas lo que se desea auditar (sobre todo a nivel base de datos) son las operaciones realizadas por usuarios fuera de las aplicaciones de gestión. Un ejemplo, seria poder capturar un UPDATE o un SELECT de un usuario utilizando herramientas como Management Studio, Excel, etc.

A tal fin y en base a las mejores prácticas es que se recomienda aplicar un filtro de que usuarios vamos a realmente auditar así luego nuestro log no se llena con eventos que quizás nunca veremos.

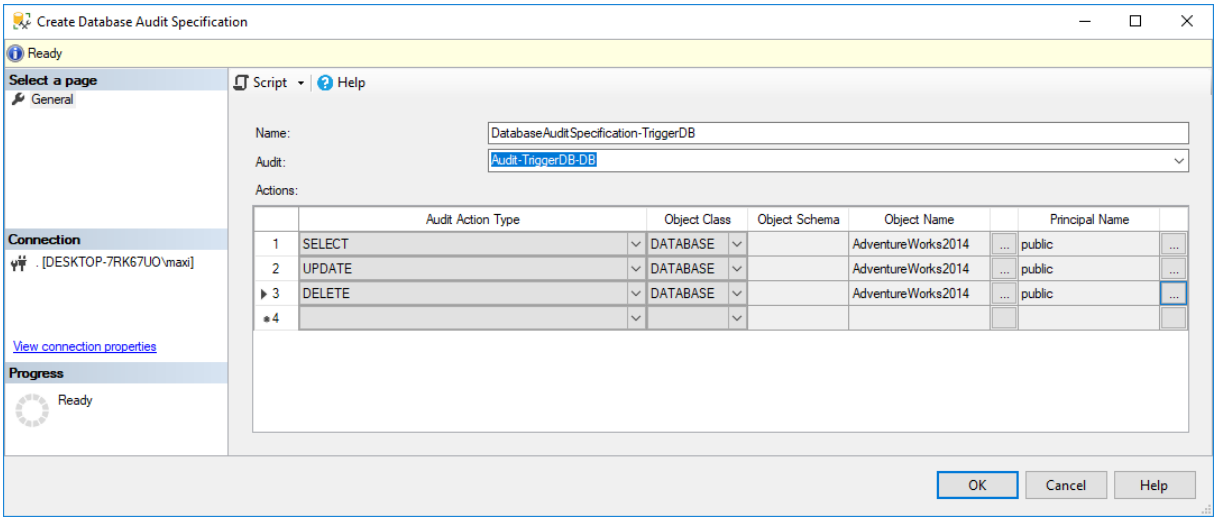
Para hacer esta operación hay dos formas posibles que luego veremos a continuación a lo largo de esta guía.

5.3 CREAR DATABASE AUDIT SPECIFICATION

Esta operación la haremos sobre la base de datos que deseamos auditar, en nuestro ejemplo usaremos “AdventureWorks2014”.



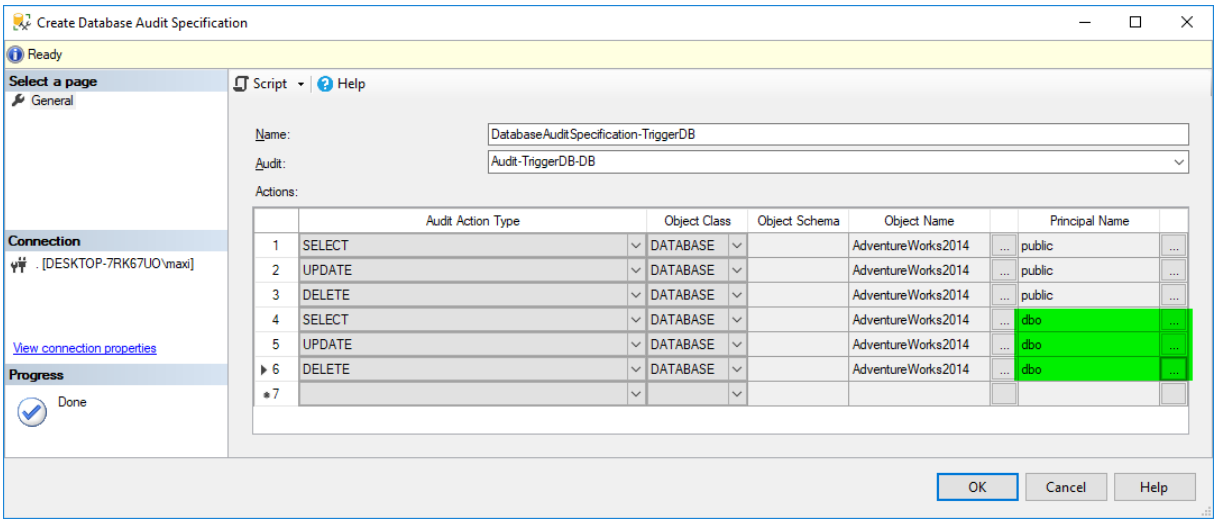
Luego las especificaciones:



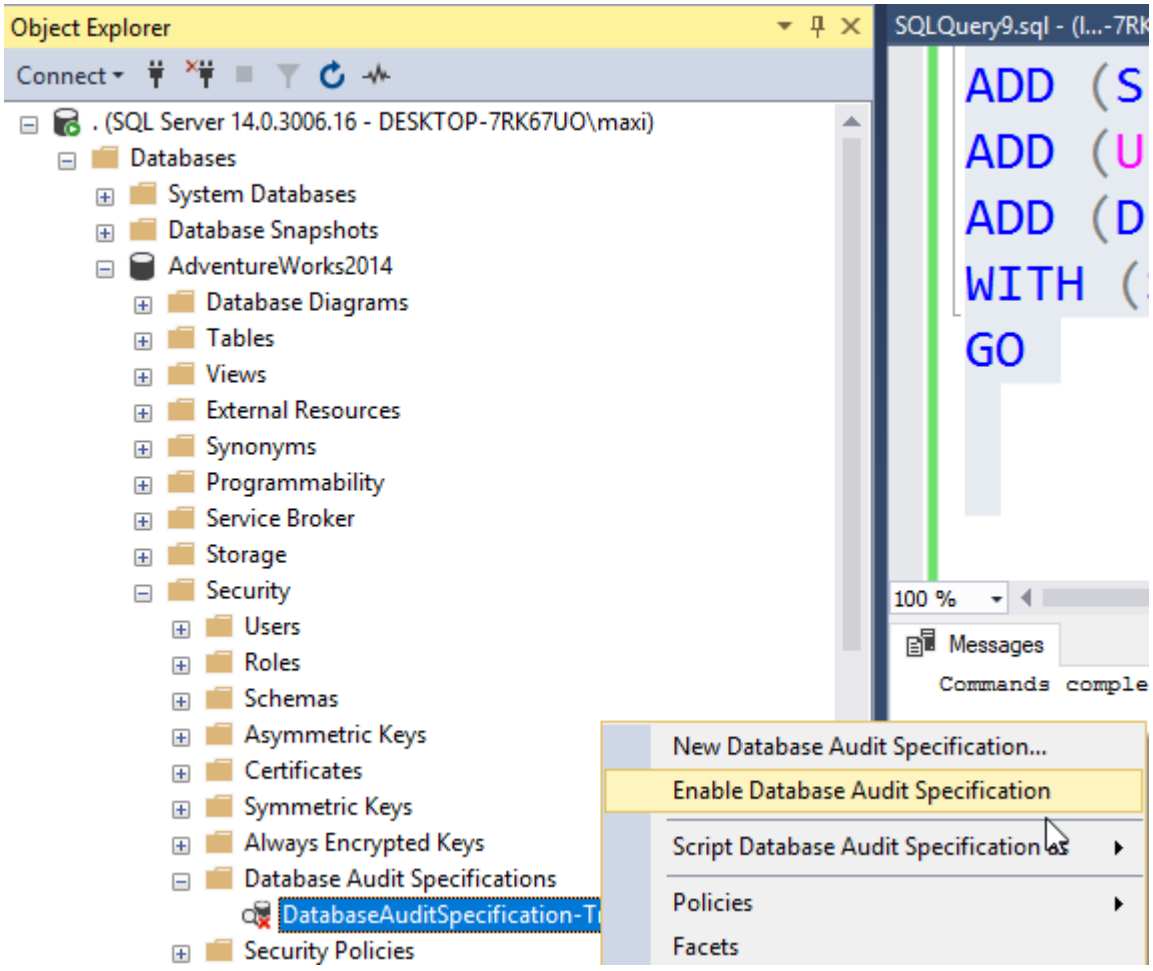
A continuación, tienes información útil:

Name(*)	Aquí ingresaremos el nombre de nuestro Database Audit Specification , en nuestro ejemplo “DatabaseAuditSpecification-TriggerDB”
Audit(*)	Aquí debemos seleccionar el Server Audit en el cual se persistirán los eventos. En nuestro ejemplo seleccionamos “Audit-TriggerDB-DB” ya que hemos divido los eventos de servidor de los de base de datos en dos Server Audit distintos.
Audit Action Type (*)	Aquí seleccionaremos los eventos a auditar. En el siguiente link se encuentra el listado de todos los eventos disponibles https://docs.microsoft.com/es-mx/sql/relational-databases/security/auditing/sql-server-audit-action-groups-and-actions
Object Class(*)	Aquí debe seleccionar el tipo de objeto a auditar donde las opciones son: ▪ **Database:** Se auditarán todos los objetos de la base de datos ▪ **Object:** Solo se auditará el objeto seleccionado (por ejemplo, una tabla en particular) ▪ **Schema:** Se auditarán los objetos que estén dentro del schema (por ejemplo, DBO)
Object Schema	Si se selecciona en object class auditar un schema, en este campo deberá indicar cuál.
Object Name	Debe indicar el nombre del objeto a auditar ya sea para Database u Object. En nuestro caso hemos seleccionado Adventureworks2014 que es el nombre de la base de datos
Principal Name(*)	Aquí debe elegir un Database Role, un usuario o un Application Role. En nuestro ejemplo hemos seleccionado al role Public , lo cual indica que auditaremos a todos los usuarios. Si desea no auditar a los usuarios de la aplicación y si a los externos, una alternativa seria crear un role en cada base de datos (por ejemplo, llamado Auditoria) y seleccionar ese role en principal name (en lugar del public) Si además desea auditar a los Sysadmin de su servidor (estos por lo general no son ni usuarios de sus bases de datos) debería agregar a los mismos eventos el role dbo (como se muestra en la figura siguiente)

A continuación tienes la especificación con mas detalle:



También es necesario que lo habilites:



A continuación, tienes el código:

```
USE [AdventureWorks2014]
GO

CREATE DATABASE AUDIT SPECIFICATION
[DatabaseAuditSpecification-TriggerDB]
FOR SERVER AUDIT [Audit-TriggerDB-DB]
ADD (SELECT ON DATABASE::[AdventureWorks2014] BY [public]),
ADD (UPDATE ON DATABASE::[AdventureWorks2014] BY [public]),
ADD (DELETE ON DATABASE::[AdventureWorks2014] BY [public]),
ADD (SELECT ON DATABASE::[AdventureWorks2014] BY [dbo]),
ADD (UPDATE ON DATABASE::[AdventureWorks2014] BY [dbo]),
ADD (DELETE ON DATABASE::[AdventureWorks2014] BY [dbo])
WITH (STATE = ON)
GO
```

6 FILTRAR REGISTROS EN EL SERVER AUDIT

Como hemos comentado anteriormente en este documento, en la auditoria quizás no tenga sentido que se registren los eventos provenientes de las aplicaciones de gestión y si de las externas.

Al crear el Database Audit Specification hemos visto que podríamos resolver esto creando un role en la base de datos y luego asignando a los usuarios que debemos auditar en dicho role, seria técnicamente como poderlos agrupar de alguna forma.

En esta sección veremos una segunda técnica que directamente aplica al Server Audit y es la posibilidad de filtrar ahí mismo sin importar de donde se haga el evento.

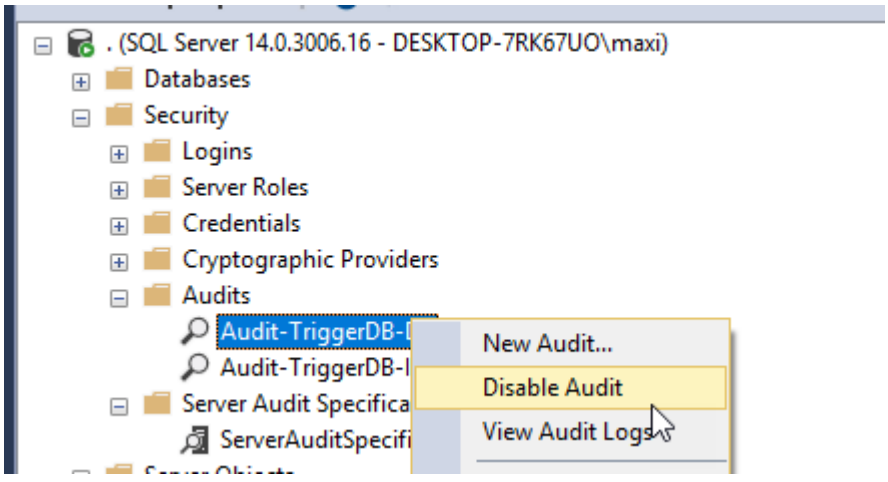
Este método si tenemos muchas bases de datos quizás es mejor que el anterior ya que nos permite mejorar la administración.

Tenga cuidado si por ejemplo desea auditar eventos en el Server Audit Specification como SUCCESFUL_LOGIN_GROUP y otros eventos más en la misma especificación, si filtra los de la aplicación estará perdiendo de datos. Imagine que desea auditar login sucess y cambios de clave y no le interesa ver en su auditoria los de login sucess que sean del login de la aplicación (le llenera el log seguramente y esa información es probable que no sea relevante para un departamento de seguridad informática).

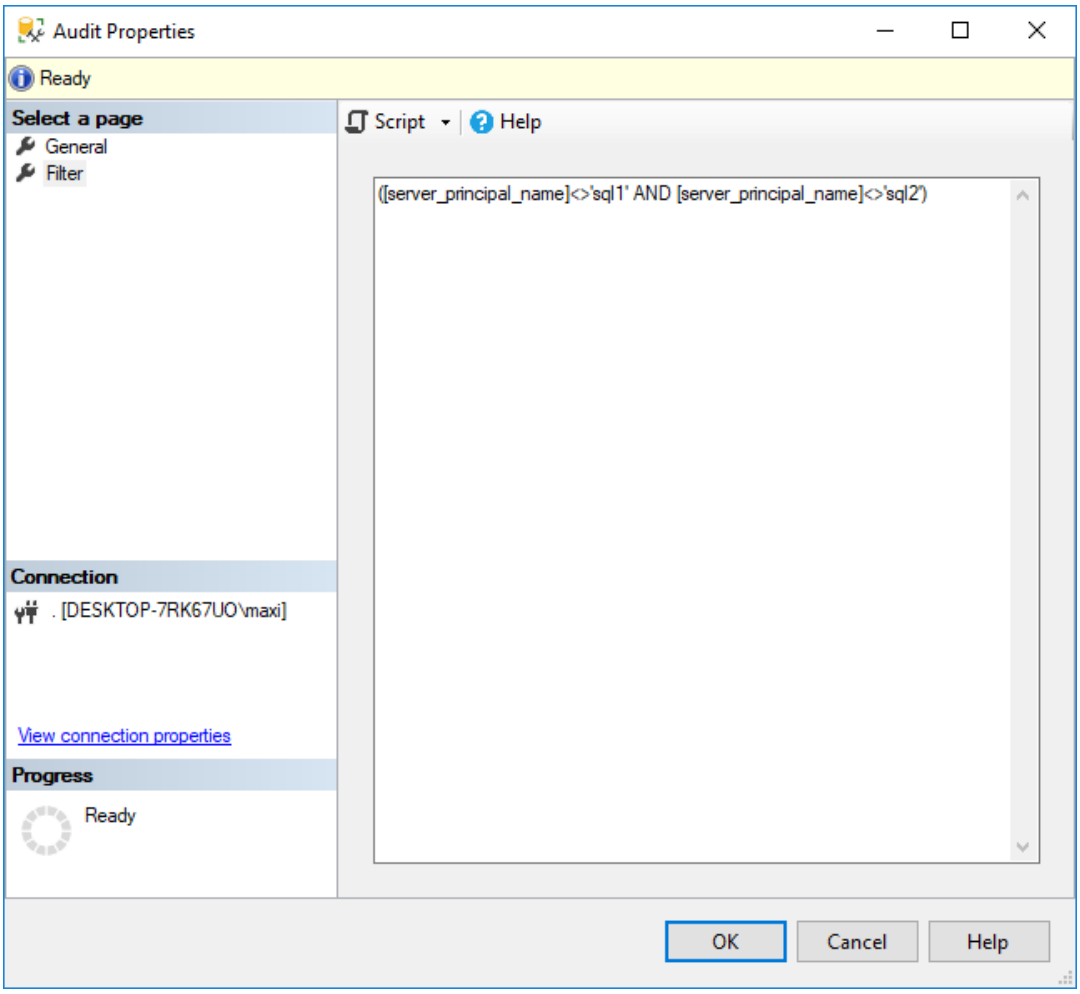
Para resolver este ultima caso lo que usted debería hacer es crear dos Server Audit distintos (uno con filtro y el otro no) y dos Server Audit Specification distintos (por ejemplo, los eventos de Successful_login_group en uno y en el otro el resto)

Para aplicar los filtros sobre un Server Audit ya existente se deben seguir los siguientes pasos

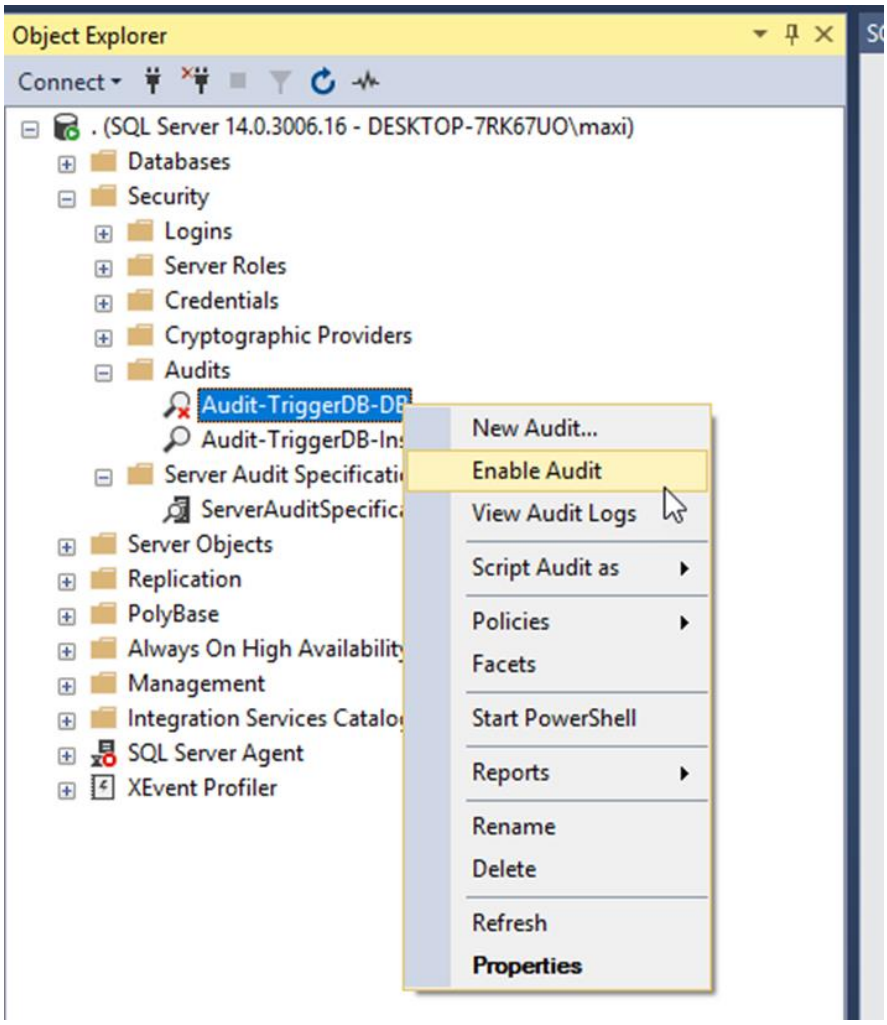
1. Poner en `disable` el Server Audit



2. Agregar al Server Audit el filtro



3. Volver a habilitar el Server Audit



A continuación, tienes el código:

```
ALTER SERVER AUDIT [Audit-TriggerDB-DB]
WITH (STATE = OFF);
GO

USE [master]
GO

ALTER SERVER AUDIT [Audit-TriggerDB-DB]
WHERE ([server_principal_name]<>'sql1'
AND [server_principal_name]<>'sql2');
GO

ALTER SERVER AUDIT [Audit-TriggerDB-DB]
WITH (STATE = ON);
GO
```

Nota:

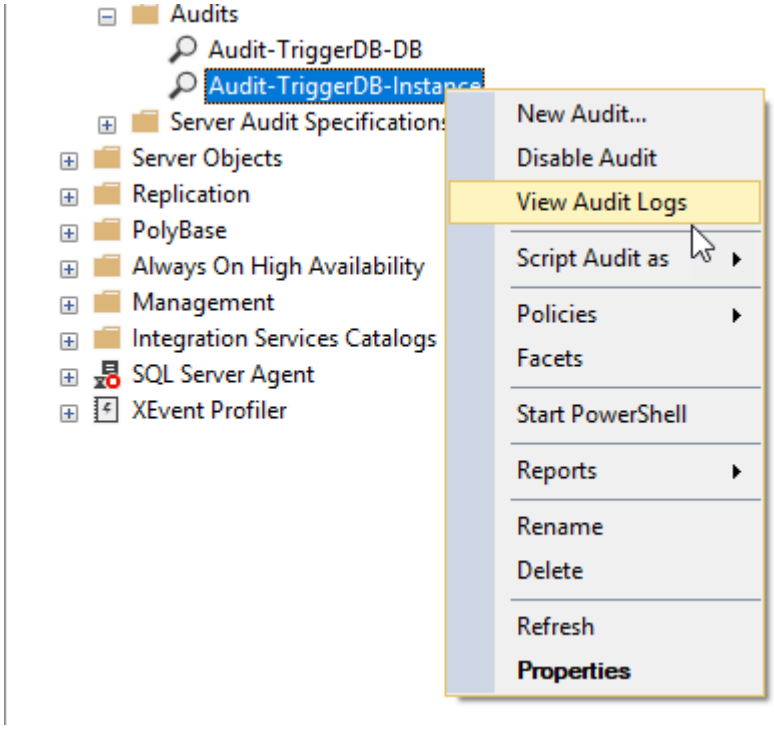
en este ejemplo hemos seleccionado el Server Audit que hemos creado para persistir los eventos de base de datos y le hemos agregado el filtro para que excluya a los login SQL1 y SQL2.

7 VER LOS RESULADOS DE LA AUDITORIA

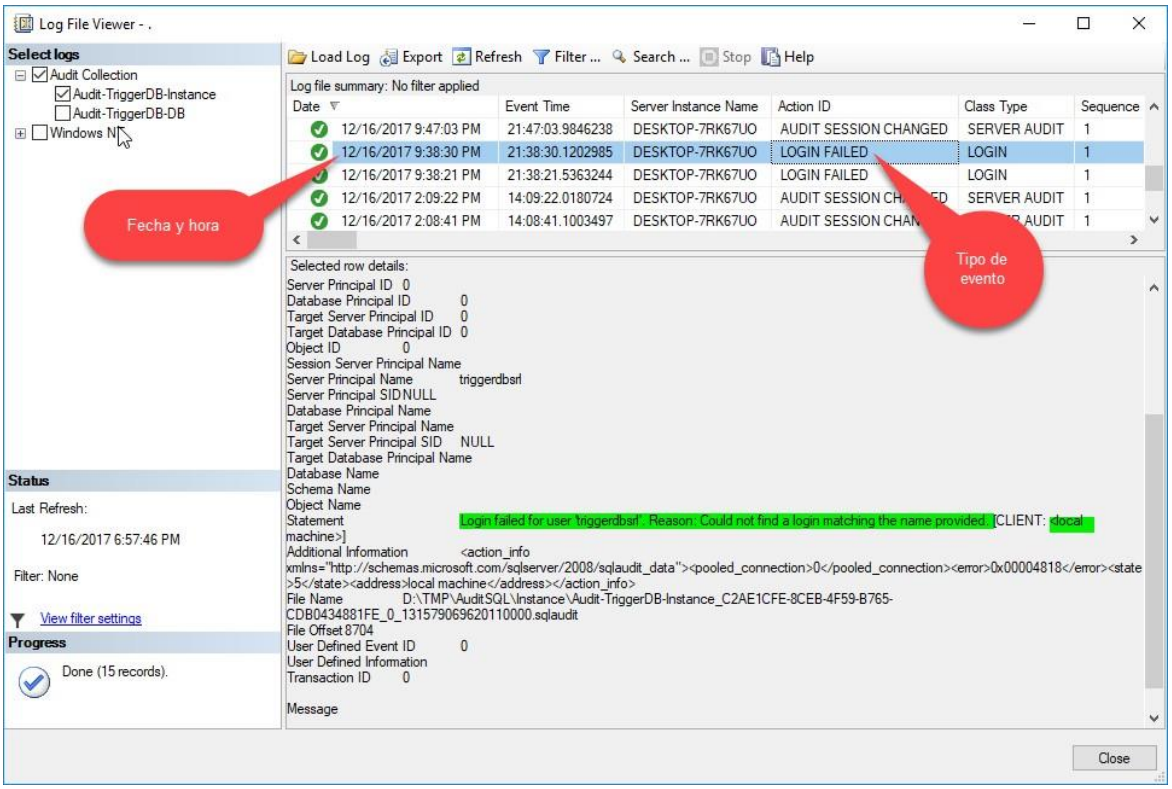
Para poder ver los distintos eventos con su correspondiente detalle existen dos alternativas que veremos a continuación.

7.1 USANDO EL SQL SERVER MANAGEMENT STUDIO (SSMS)

Desde el propio SSMS se pueden ver los registros del log, para eso lo que se debe hacer es lo siguiente sobre el Server Audit que deseamos observar:

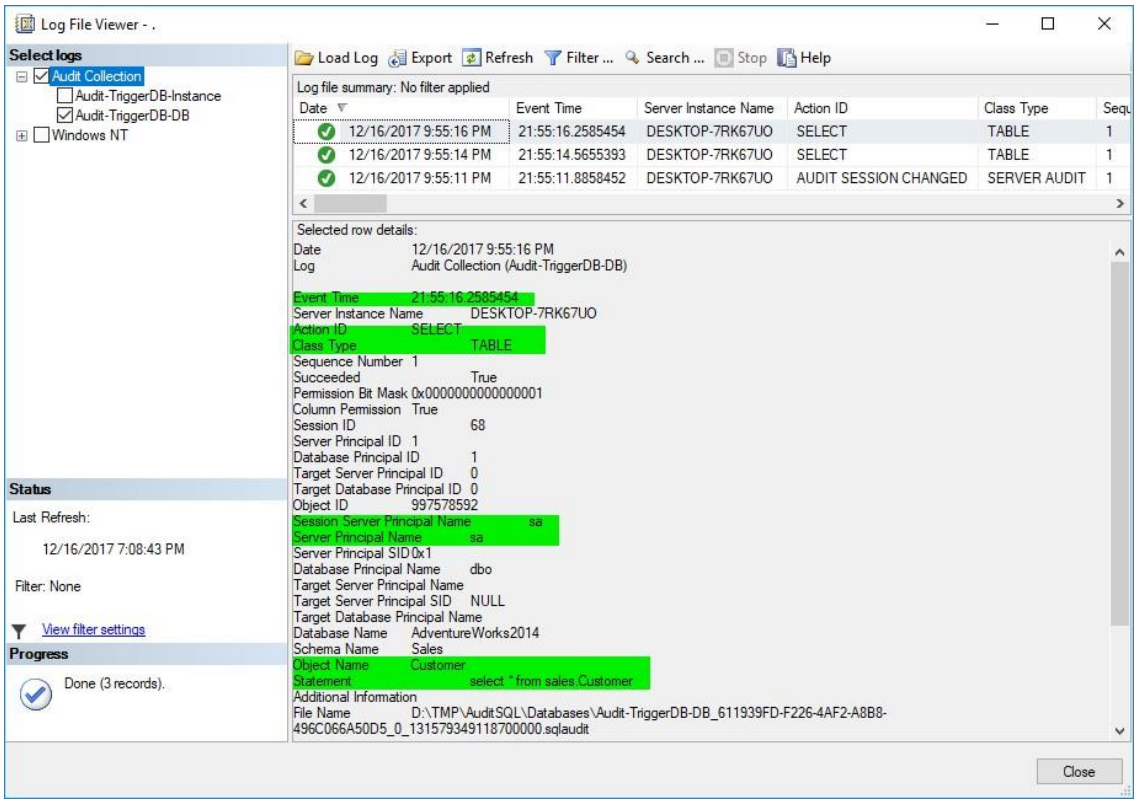


A continuación se observa los detalles:



En esta última figura podemos observar que hubo un login fail y todo el detalle del mismo (fecha y hora, tipo de acción, etc)

Si hacemos el mismo procedimiento sobre el server Audit que aloja los eventos de base de datos veremos los mismos atributos, pero obviamente con otros datos



7.2 USANDO CODIGO TSQL

Otra alternativa mucho más completa y customizada es poder usar código TSQL, esto nos permitirá entre varias cosas por ejemplo integrar o armar informes en herramientas como PowerBI, Excel, reporting Services, etc.

Para poder usar esta opción SQL Server dispone de una función llamada sys.fn_get_audit_file, para más información puede consultar el siguiente enlace: https://docs.microsoft.com/en-us/sql/relational-databases/system-functions/sys-fn-get-audit-file-transact-sql.

Con la cual podemos leer los archivos de auditoria y que el resultado sea una tabla para luego verlo o integrarlo con otras soluciones.

Con esta función y las vistas de SQL Server correspondientes a Audit se puede buscar toda la información necesaria

sys.server_audits	Contiene una fila para cada auditoría de SQL Server de una instancia de servidor https://docs.microsoft.com/en-us/sql/relational-databases/system-catalog-views/sys-server-audits-transact-sql
sys.server_file_audits	Contiene información adicional sobre el tipo de auditoría de archivos en un SQL Server https://docs.microsoft.com/es-mx/sql/relational-databases/system-catalog-views/sys-server-file- audits-transact-sql
sys.server_file_audits	Contiene información adicional sobre el tipo de auditoría de archivos en un SQL Server https://docs.microsoft.com/es-mx/sql/relational-databases/system-catalog-views/sys-server-file- audits-transact-sql
sys.server_audit_specifications	Contiene información sobre las especificaciones de auditoría de servidor de una auditoría de SQL Server https://docs.microsoft.com/es-mx/sql/relational-databases/system-catalog-views/sys-server-audit- specifications-transact-sql
sys.server_audit_specification_details	Contiene información sobre los detalles de especificación de auditoría del servidor (acciones) https://docs.microsoft.com/es-es/sql/relational-databases/system-catalog-views/sys-server-audit- specification-details-transact-sql

sys.database_audit_specifications	Contiene información sobre las especificaciones de auditoría de base de datos https://docs.microsoft.com/es-es/sql/relational-databases/system-catalog-views/sys-database- audit-specifications-transact-sql
sys.database_audit_specification_details	Contiene información sobre las especificaciones de auditoría de base de datos en una auditoría de SQL Server de una instancia de servidor para todas las bases de datos https://docs.microsoft.com/es-es/sql/relational-databases/system-catalog-views/sys-database- audit-specification-details-transact-sql
sys.dm_server_audit_status	Devuelve una fila para cada auditoría de servidor que indica el estado actual de la misma https://docs.microsoft.com/es-es/sql/relational-databases/system-dynamic-management-views/sys- dm-server-audit-status-transact-sql
sys.dm_audit_actions	Devuelve una fila por cada acción de auditoría sobre la que se puede guardar información en el registro de auditoría y por cada grupo de acciones de auditoría que se puede configurar como parte de SQL Server Audit https://docs.microsoft.com/es-es/sql/relational-databases/system-dynamic-management-views/sys- dm-audit-actions-transact-sql
sys.dm_audit_class_type_map	Devuelve una tabla que asigna el campo class_type del registro de auditoría al campo class_desc en sys.dm_audit_actions https://docs.microsoft.com/es-es/sql/relational-databases/system-dynamic-management-views/sys- dm-audit-class-type-map-transact-sql

El siguiente código TSQL buscara todos los eventos de los archivos existentes para la auditoria “Audit-TriggerDB-Instance”:

```
DECLARE @PATH VARCHAR(1024)
SELECT @PATH = LOG_FILE_PATH + '*.*'
FROM sys.server_file_audits
WHERE name = 'Audit-TriggerDB-Instance'

SELECT A.NAME,
A.class_desc, A.parent_class_desc, A.covering_parent_action_name,
F.*
FROM sys.fn_get_audit_file (@PATH,default,default) as F
left join sys.dm_audit_actions A
on F.action_id = A.action_id
ORDER BY EVENT_TIME DESC;
GO
```

Results Messages

	NAME	class_desc	parent_class_desc	covering_parent_action_name	event_time	sequence_number	action_id	succeeded	permission_bitmask
1	LOGIN FAILED	LOGIN	SERVER	FAILED_LOGIN_GROUP	2017-12-16 21:53:29.8367338	1	LGIF	0	0x0000000000000000
2	LOGIN FAILED	LOGIN	SERVER	FAILED_LOGIN_GROUP	2017-12-16 21:53:29.8327310	1	LGIF	0	0x0000000000000000
3	LOGIN FAILED	LOGIN	SERVER	FAILED_LOGIN_GROUP	2017-12-16 21:53:29.8327310	1	LGIF	0	0x0000000000000000
4	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:53:16.1506885	1	AUSC	1	0x0000000000000000
5	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:53:00.5326848	1	AUSC	1	0x0000000000000000
6	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:53:00.5326848	1	AUSC	1	0x0000000000000000
7	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:47:54.1257572	1	AUSC	1	0x0000000000000000
8	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:47:03.9966316	1	AUSC	1	0x0000000000000000
9	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 21:47:03.9846238	1	AUSC	1	0x0000000000000000
10	LOGIN FAILED	LOGIN	SERVER	FAILED_LOGIN_GROUP	2017-12-16 21:38:30.1202985	1	LGIF	0	0x0000000000000000
11	LOGIN FAILED	LOGIN	SERVER	FAILED_LOGIN_GROUP	2017-12-16 21:38:21.5363244	1	LGIF	0	0x0000000000000000
12	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 14:09:22.0180724	1	AUSC	1	0x0000000000000000
13	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 14:08:41.1003497	1	AUSC	1	0x0000000000000000
14	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 14:08:41.0963504	1	AUSC	1	0x0000000000000000
15	AUDIT SESSION CHANGED	SERVER AUDIT	SERVER	NULL	2017-12-16 14:08:37.7422663	1	AUSC	1	0x0000000000000000

8 PERFORMANCE Y CONCLUSIONES

La utilización de eventos y asincronismo hacen que las implementaciones de las auditorias nativas no tengan impacto en la performance de nuestro motor como si suele suceder con otras técnicas como por ejemplo el uso de trigger.

Las auditorias están disponibles desde SQL Server 2008 lo cual la hacen una funcionalidad madura.

Si bien se pueden usar otros métodos de auditoria (extend Events y profiler entre otros) las auditorias nativas son robustas y contienen todo lo necesario para una implementación adecuada.

9 CURSOS RELACIONADOS

https://www.ceps.uni.edu.pe/


SQL SERVER
INTELIGENCIA DE
NEGOCIOS
NIVEL IV


Fundamentos
de Programación


PROGRAMACIÓN
ORIENTADA A OBJETOS
CON JAVA


CLIENTE - SERVIDOR


DESARROLLO WEB


FACEBOOK
PARA NEGOCIO