Microsoft

Security - General

Module 4

Learning Units covered in this Module

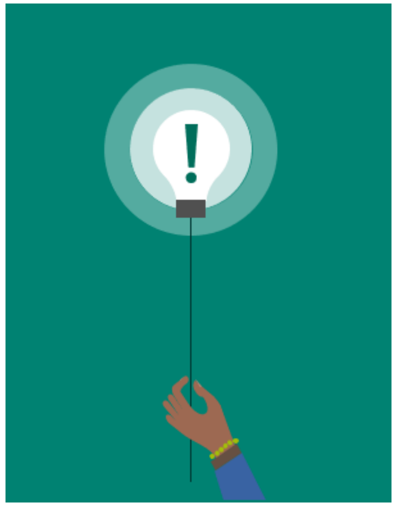
- Lesson 1: Authentication and Authorization
- Lesson 2: Row Level Security
- Lesson 3: Dynamic Data Masking
- Lesson 4: SQL Server Audit

Lesson 1: Authentication and Authorization

Objectives

After completing this learning, you will be able to:

- Understand the difference between Authentication and Authorization
- Understand the difference between Principals and Securables
- Understand how to create logins and users
- Understand how to assign permissions to objects.
- Understand the concept of SQL Server schemas

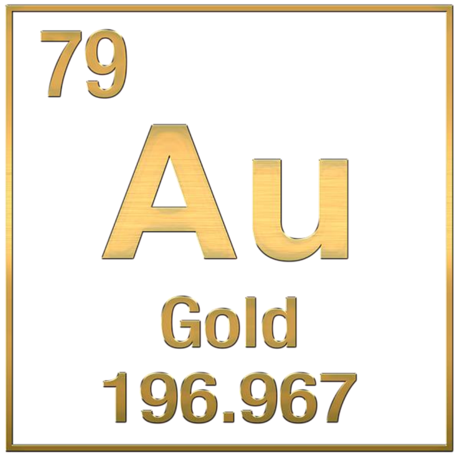


The Security Gold Standard

AUTHENTICATION – Verifies who you are

AUTHORIZATION – Assigns what you can do

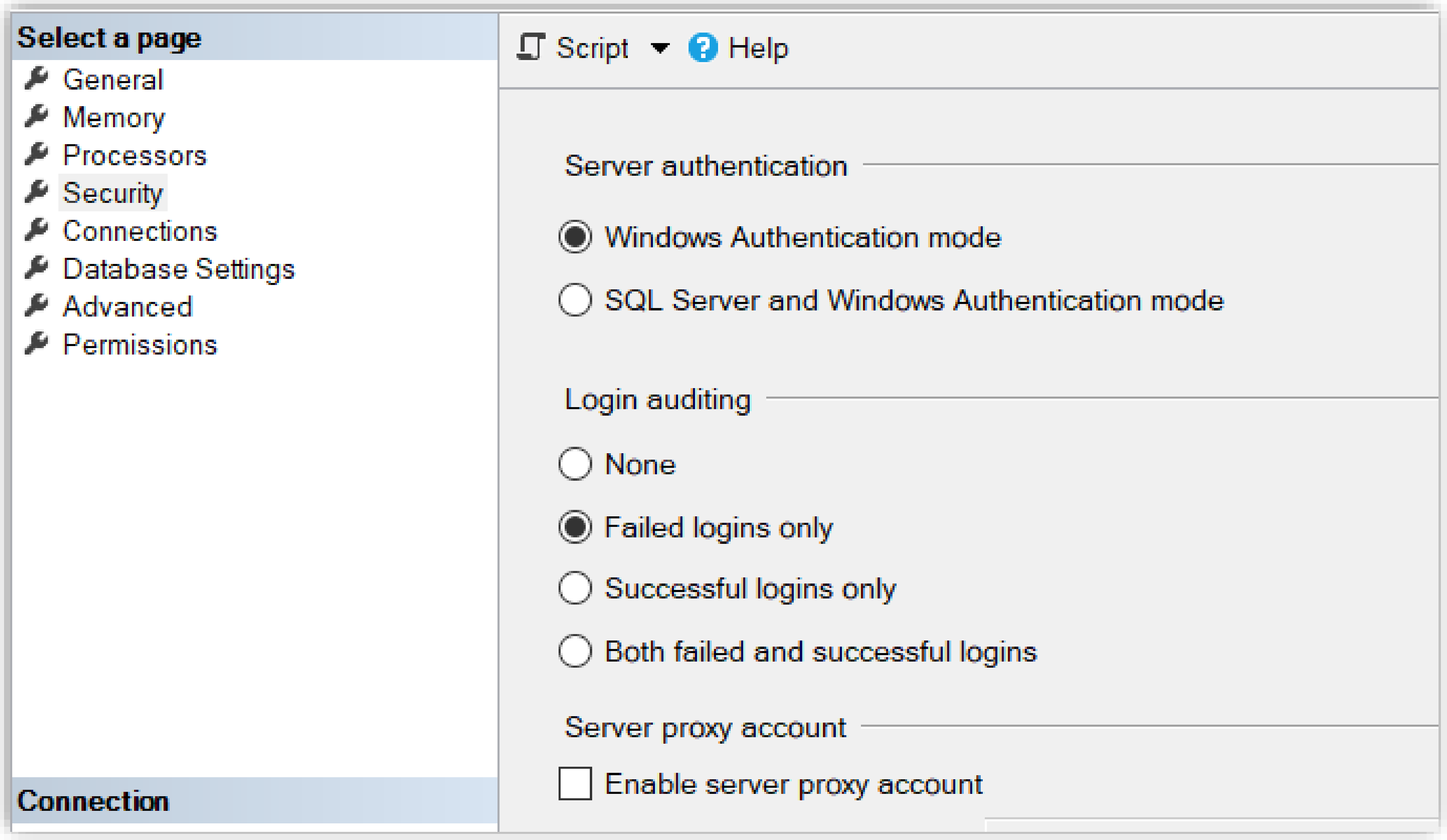
AUDITING – Monitors what you did



Authentication Types

Windows Authentication	SQL Authentication
• SQL Server validates credentials using Active Directory and then verifies if it has permissions to connect.	• SQL Server validates the password against a hash stored in master and then verifies if it has permissions to connect.

Server Authentication



Select a page
- General
- Memory
- Processors
- Security
- Connections
- Database Settings
- Advanced
- Permissions

Connection

Script ▾ Help

Server authentication

- [✓ Windows Authentication mode]
- SQL Server and Windows Authentication mode

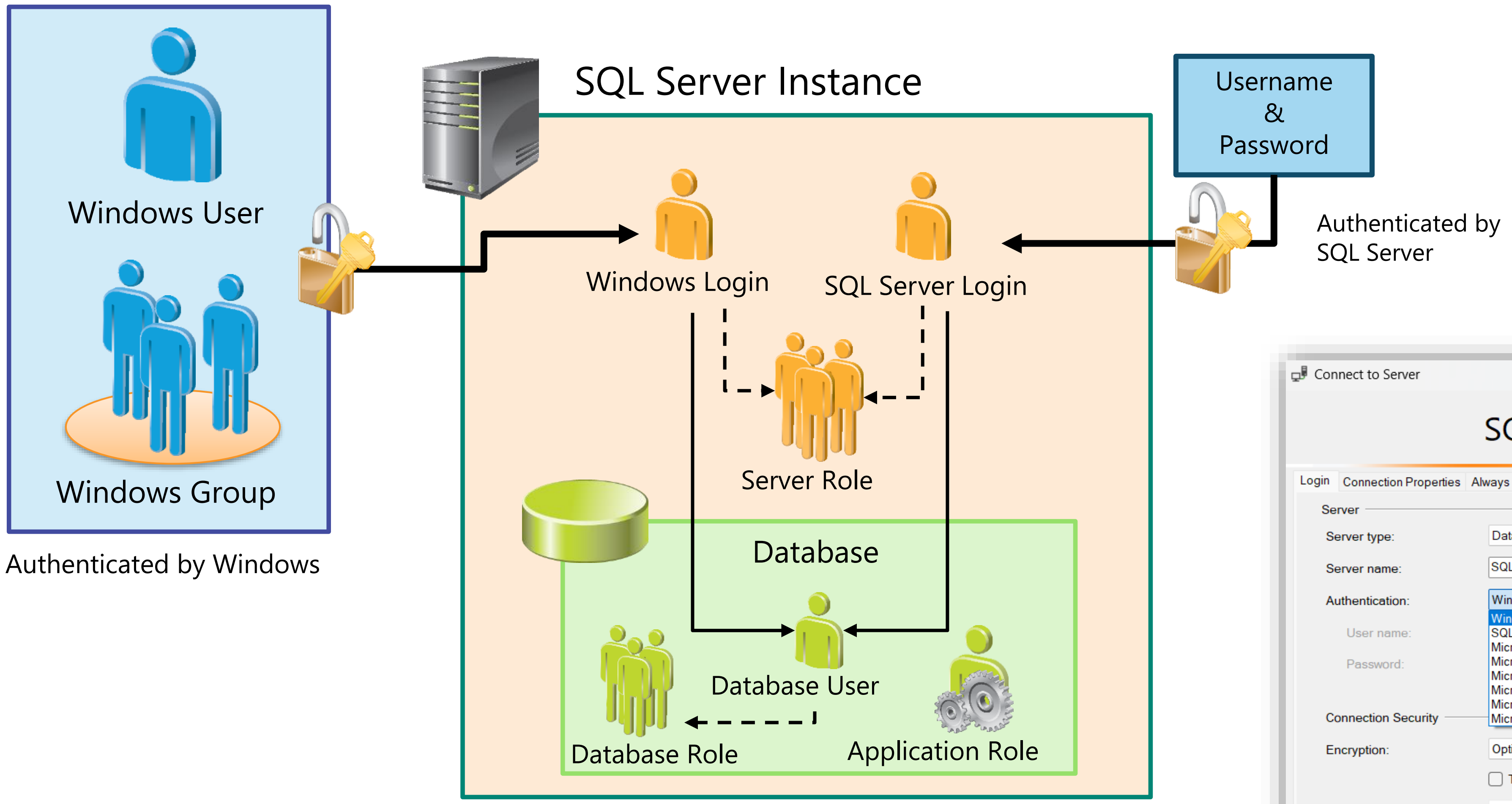
Login auditing

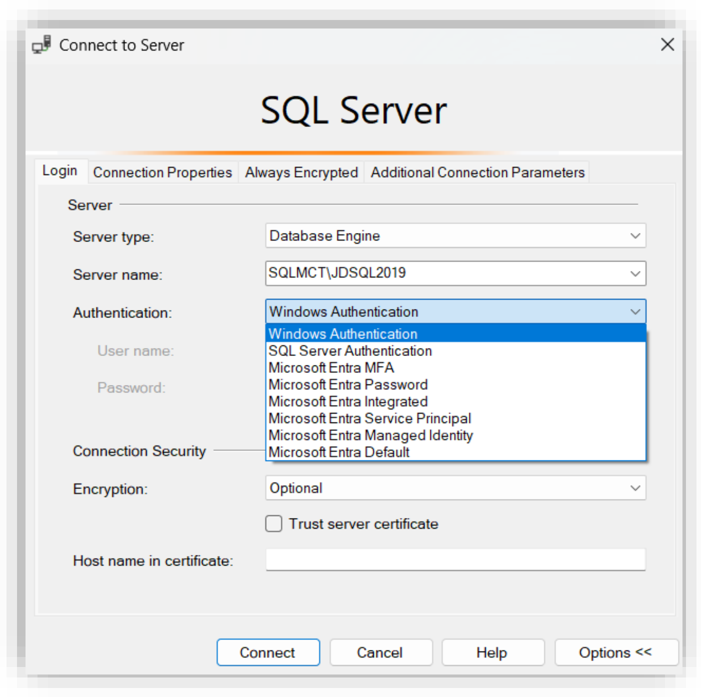
- None
- [✓ Failed logins only]
- Successful logins only
- Both failed and successful logins

Server proxy account

- [ ] Enable server proxy account

Security Principals




Connect to Server

SQL Server

Login | Connection Properties | Always Encrypted | Additional Connection Parameters

Server

Server type:	Database Engine
Server name:	SQLMCT\JDSQL2019
Authentication:	Windows Authentication
User name:	
Password:	

Windows Authentication
SQL Server Authentication
Microsoft Entra MFA
Microsoft Entra Password
Microsoft Entra Integrated
Microsoft Entra Service Principal
Microsoft Entra Managed Identity
Microsoft Entra Default

Connection Security

Encryption:	Optional
	☐ Trust server certificate
Host name in certificate:	

Connect | Cancel | Help | Options <<

Creating Logins

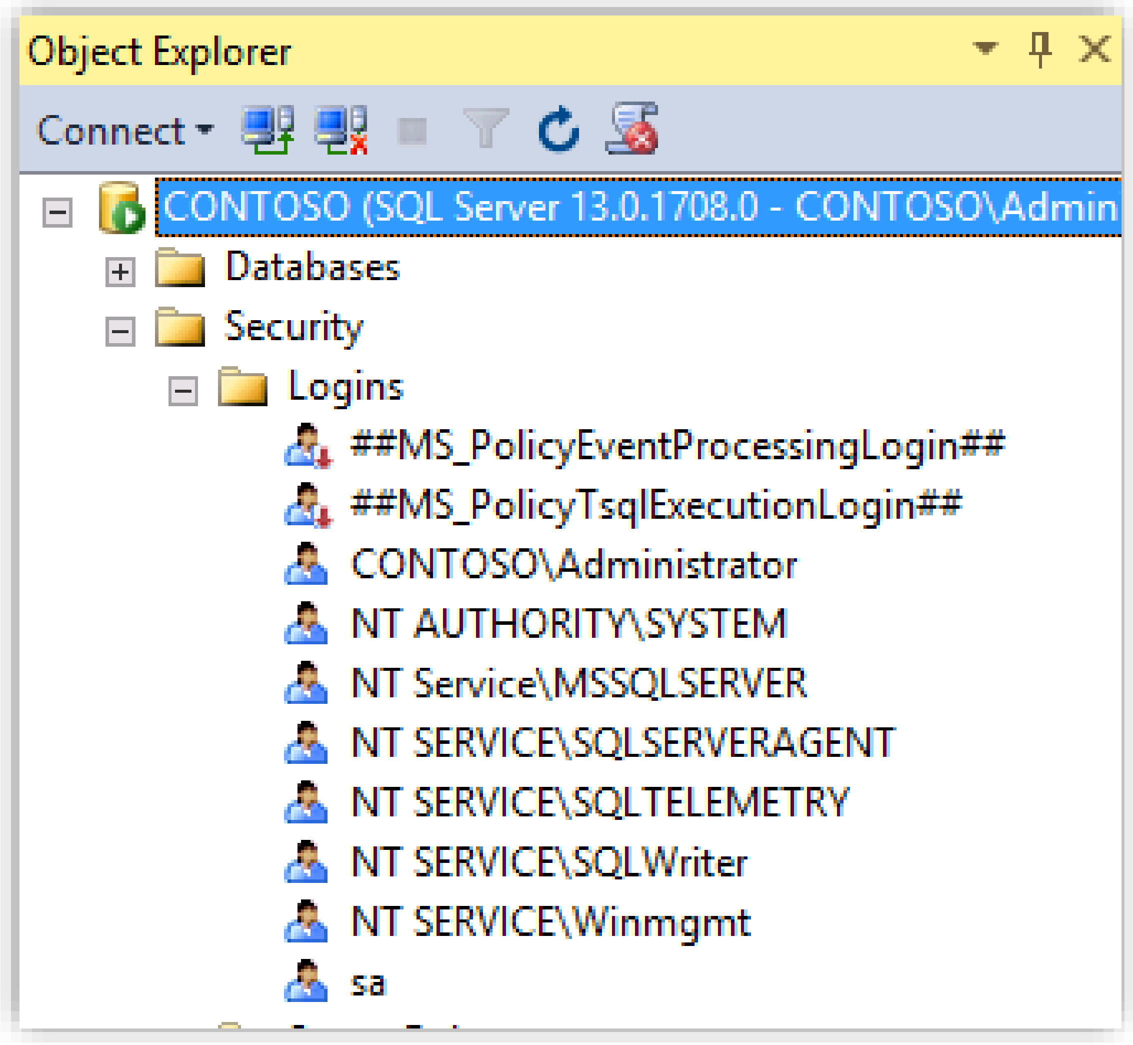
Allows connection to a SQL Server Instance

Two type of logins:

- SQL Login
- Windows Login

Can be created by:

- CREATE LOGIN statement in T-SQL
- SQL Server Management Studio



CONTOSO (SQL Server 13.0.1708.0 - CONTOSO\Admin
Databases
Security
Logins
##MS_PolicyEventProcessingLogin##
##MS_PolicyTsqlExecutionLogin##
CONTOSO\Administrator
NT AUTHORITY\SYSTEM
NT Service\MSSQLSERVER
NT SERVICE\SQLSERVERAGENT
NT SERVICE\SQLTELEMETRY
NT SERVICE\SQLWriter
NT SERVICE\Winmgmt
sa

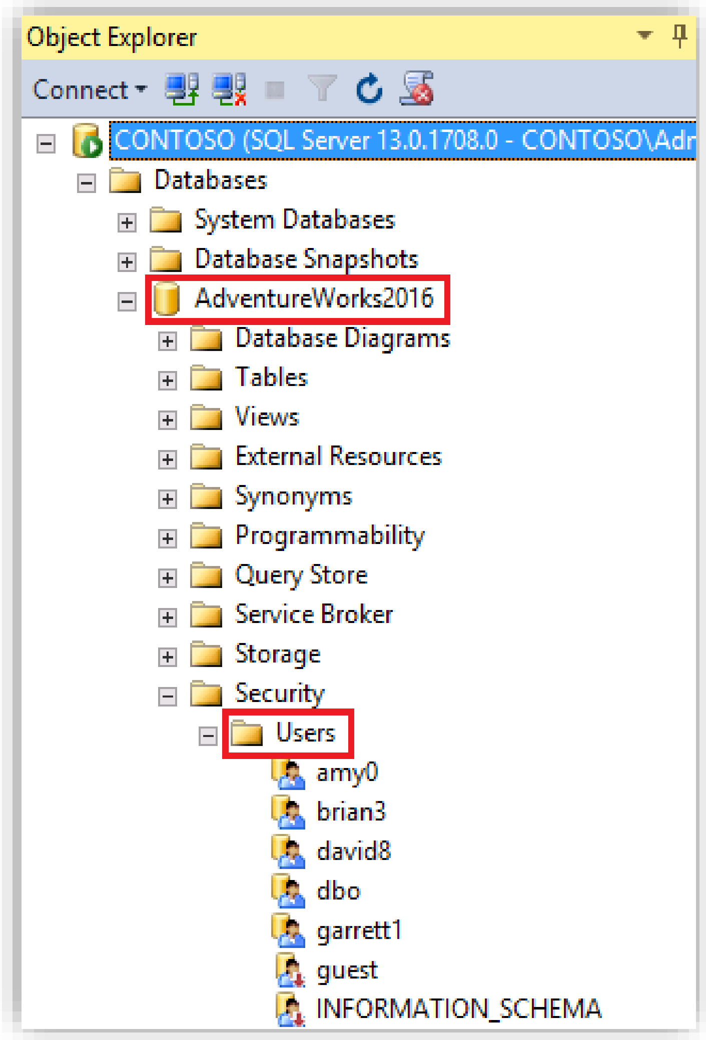
Creating Users

Allow access to a database

Specific to a single database

Type of users:

- Windows user
- SQL User with Password
- SQL User with Login
- SQL User without Password
- User mapped to a certificate
- User mapped to an asymmetric key



Object Explorer

Connect

```
CONTOSO (SQL Server 13.0.1708.0 - CONTOSO\Adr
  Databases
    System Databases
    Database Snapshots
    AdventureWorks2016
      Database Diagrams
      Tables
      Views
      External Resources
      Synonyms
      Programmability
      Query Store
      Service Broker
      Storage
      Security
        Users
          amy0
          brian3
          david8
          dbo
          garrett1
          guest
          INFORMATION_SCHEMA
```

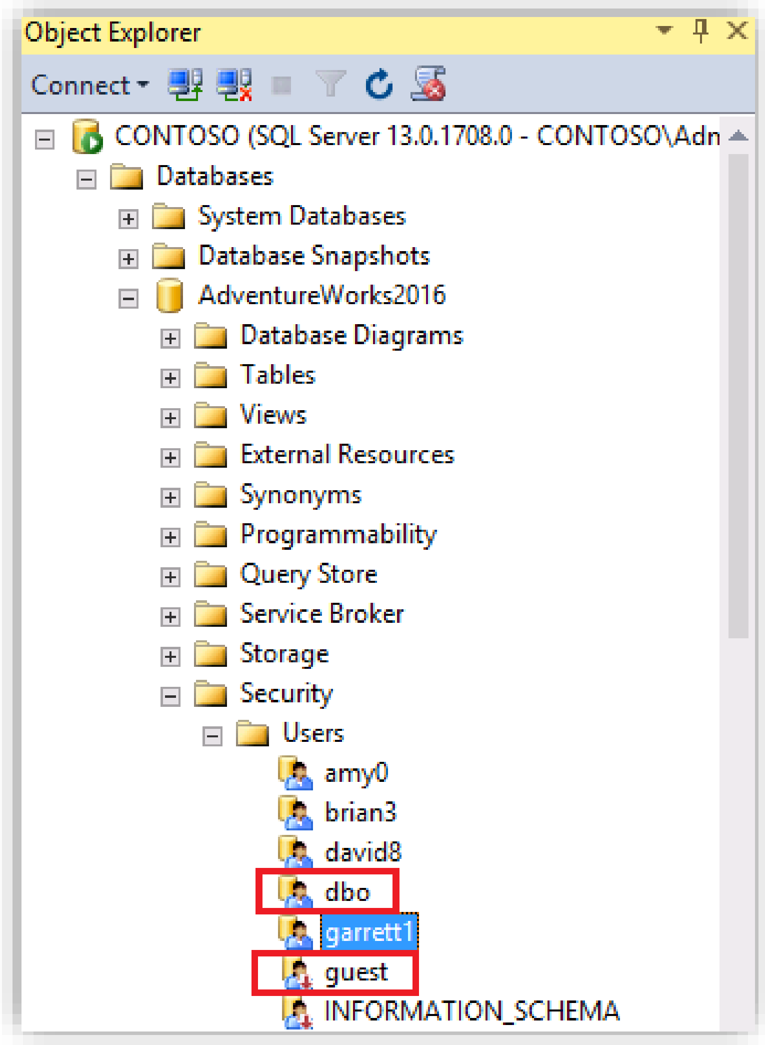
DBO and Guest User

DBO

- Performs all activities in the database
- Members of sysadmin role, SA login, and database owner are mapped to DBO.
- Cannot be deleted

Guest

- Allows logins without user accounts to access database
- Disabled by default in user databases
- Cannot be dropped but you can prevent it from accessing a database
- Must NOT be disabled in master and tempdb



Roles

Server Roles

- Fixed server roles
- User-defined server roles

Database Roles

- Fixed database roles
- User-defined database roles

Application roles

- Assign rights to applications instead of users

Fixed Server Level Roles and Permissions

Role	Description	Server-level Permission
sysadmin	Perform any activity	CONTROL SERVER (with GRANT option)
dbcreator	Create and alter databases	ALTER ANY DATABASE
diskadmin	Manage disk files	ALTER RESOURCES
serveradmin	Configure server-wide settings	ALTER ANY ENDPOINT, ALTER RESOURCES, ALTER SERVER STATE, ALTER SETTINGS, SHUTDOWN, VIEW SERVER STATE
securityadmin	Manage and audit server logins	ALTER ANY LOGIN
processadmin	Manage SQL Server processes	ALTER ANY CONNECTION ALTER SERVER STATE
bulkadmin	Run the BULK INSERT statement	ADMINISTER BULK OPERATIONS
setupadmin	Configure replication and linked servers	ALTER ANY LINKED SERVER

New Server Level Roles introduced in SQL Server 2022

Server-level role	Description
##MS_DatabaseConnector##	Connect to any database without requiring a database User-account.
##MS_LoginManager##	Create, delete and modify logins. Cannot GRANT.
##MS_DatabaseManager##	Create and delete databases.
##MS_ServerStateManager##	Same as the ##MS_ServerStateReader## role but also has the **ALTER SERVER STATE** permission.
##MS_ServerStateReader##	Read all dynamic management views (DMVs) and functions that are covered by **VIEW SERVER STATE**.
##MS_ServerPerformanceStateReader##	Read all dynamic management views (DMVs) and functions that are covered by **VIEW SERVER PERFORMANCE STATE**
##MS_ServerSecurityStateReader##	Read all dynamic management views (DMVs) and functions that are covered by **VIEW SERVER SECURITY STATE**
##MS_DefinitionReader##	Read all catalog views that are covered by **VIEW ANY DEFINITION**
##MS_PerformanceDefinitionReader##	Read all catalog views that are covered by **VIEW ANY PERFORMANCE DEFINITION.**
##MS_SecurityDefinitionReader##	Read all catalog views that are covered by **VIEW ANY SECURITY DEFINITION**.

Public Role

Public is a special role that is at the server and database level.

Every SQL Server login and user belongs to the Public role

Care must be taken when granting permissions to Public server role especially when granting server-level **permissions.**

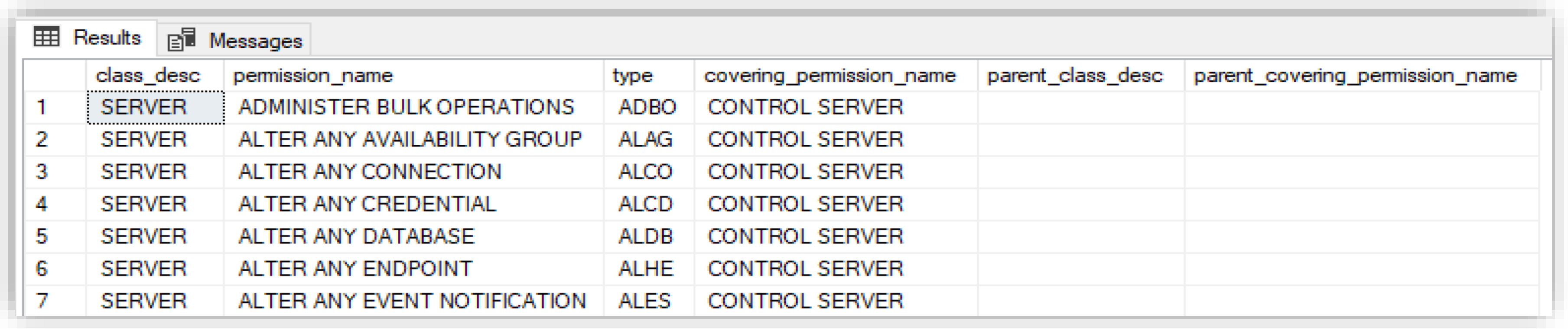
Fixed Database Level Roles and Permissions

Role	Description
db_owner	Perform any configuration and maintenance activities on the DB and can drop it
db_securityadmin	Modify role membership and manage permissions
db_accessadmin	Add or remove access to the DB for logins
db_backupoperator	Back up the DB
db_ddladmin	Run any DDL command in the DB
db_datawriter	Add, delete, or change data in all user tables
db_datareader	Read all data from all user tables
db_denydatawriter	Cannot add, delete, or change data in user tables
db_denydatareader	Cannot read any data in user tables

Listing Built-in Server and Database Permissions

SELECT * FROM sys.fn_builtin_permissions('SERVER’)

ORDER BY permission_name;

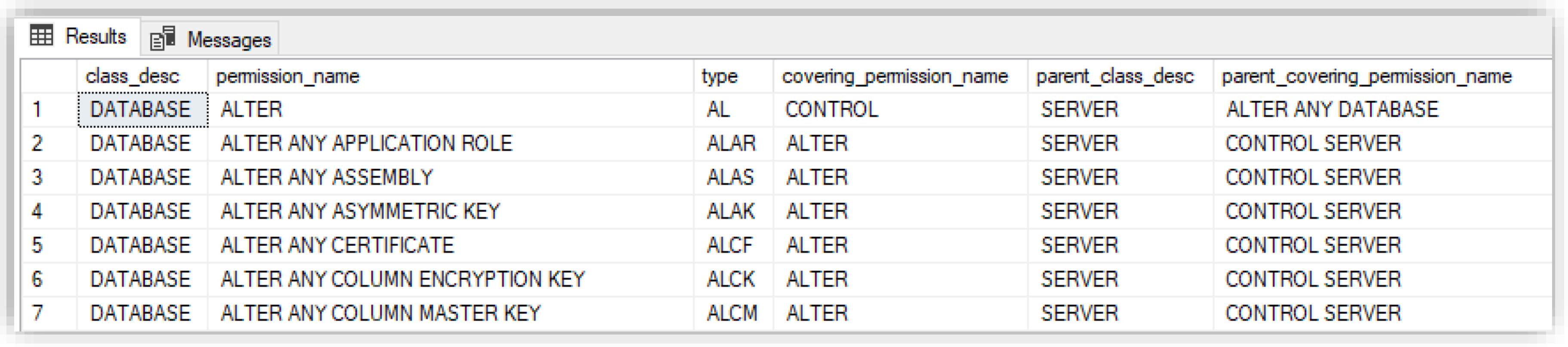


Results | Messages

	class_desc	permission_name	type	covering_permission_name	parent_class_desc	parent_covering_permission_name
1	SERVER	ADMINISTER BULK OPERATIONS	ADBO	CONTROL SERVER		
2	SERVER	ALTER ANY AVAILABILITY GROUP	ALAG	CONTROL SERVER		
3	SERVER	ALTER ANY CONNECTION	ALCO	CONTROL SERVER		
4	SERVER	ALTER ANY CREDENTIAL	ALCD	CONTROL SERVER		
5	SERVER	ALTER ANY DATABASE	ALDB	CONTROL SERVER		
6	SERVER	ALTER ANY ENDPOINT	ALHE	CONTROL SERVER		
7	SERVER	ALTER ANY EVENT NOTIFICATION	ALES	CONTROL SERVER		

SELECT * FROM sys.fn_builtin_permissions(‘Database’)

ORDER BY permission_name;



Results | Messages

	class_desc	permission_name	type	covering_permission_name	parent_class_desc	parent_covering_permission_name
1	DATABASE	ALTER	AL	CONTROL	SERVER	ALTER ANY DATABASE
2	DATABASE	ALTER ANY APPLICATION ROLE	ALAR	ALTER	SERVER	CONTROL SERVER
3	DATABASE	ALTER ANY ASSEMBLY	ALAS	ALTER	SERVER	CONTROL SERVER
4	DATABASE	ALTER ANY ASYMMETRIC KEY	ALAK	ALTER	SERVER	CONTROL SERVER
5	DATABASE	ALTER ANY CERTIFICATE	ALCF	ALTER	SERVER	CONTROL SERVER
6	DATABASE	ALTER ANY COLUMN ENCRYPTION KEY	ALCK	ALTER	SERVER	CONTROL SERVER
7	DATABASE	ALTER ANY COLUMN MASTER KEY	ALCM	ALTER	SERVER	CONTROL SERVER

Authorization

- Process by which SQL server decides whether a given principal can access a resource
- Allows granting the specific permissions required rather than granting membership in a fixed role
- Provides information and metadata of a securable only to those principals who have permission to access the securable
- Allows creating custom permission sets
- Works on the principle of *least privilege*

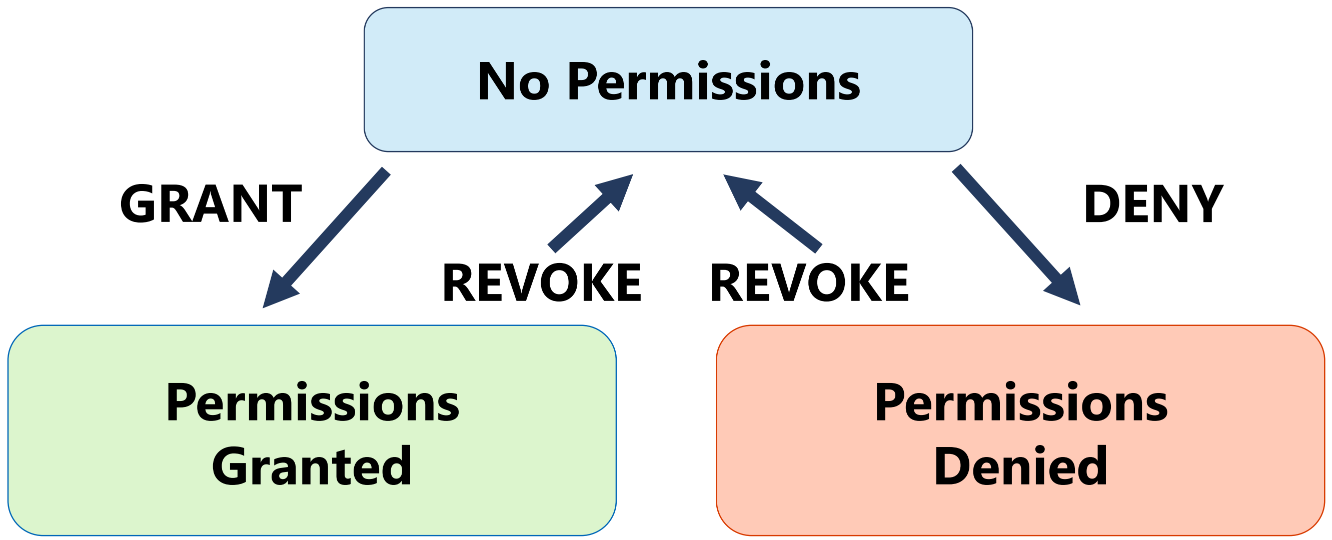
Assigning Permissions to Accounts

GRANT is used to assign a permission

DENY is used to explicitly deny a permission

- Used where permissions inherited through group or role membership
- Should only be used in exceptional circumstances

REVOKE removes either a GRANT or a DENY



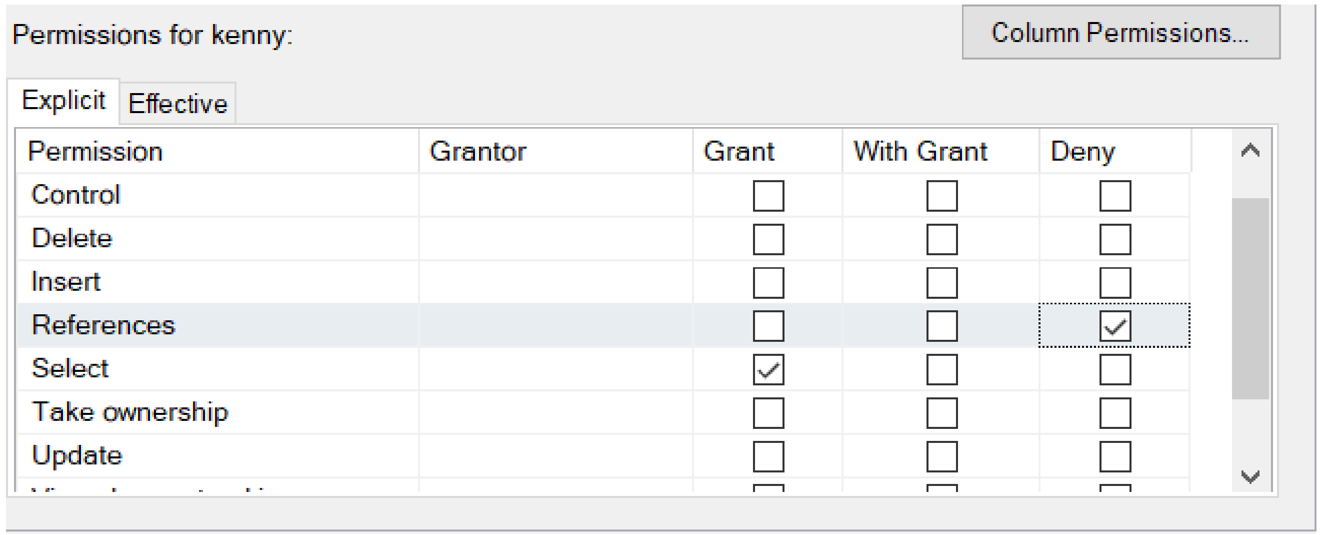
Assigning Permissions to Tables and Views

Grant with Grant allows the user to assign that permission.

Tables and Views can be assigned the same permissions.

Permissions for SELECT, UPDATE, and REFERENCES can also be set at the column level.

Select the Effective Tab to see what permissions have been granted.



Permissions for kenny:

Column Permissions...

Explicit | Effective

Permission	Grantor	Grant	With Grant	Deny
Control		☐	☐	☐
Delete		☐	☐	☐
Insert		☐	☐	☐
References		☐	☐	✓
Select		✓	☐	☐
Take ownership		☐	☐	☐
Update		☐	☐	☐

Security with Schemas

- FQN has the form: ***server.database.schema.object***
- In a database, all objects are created within a schema (dbo is default).
- Allow their owners full control over objects within the schema
- Permissions can be granted at the schema level.
- Can contain objects owned by multiple database users
- Can be owned by any database principal

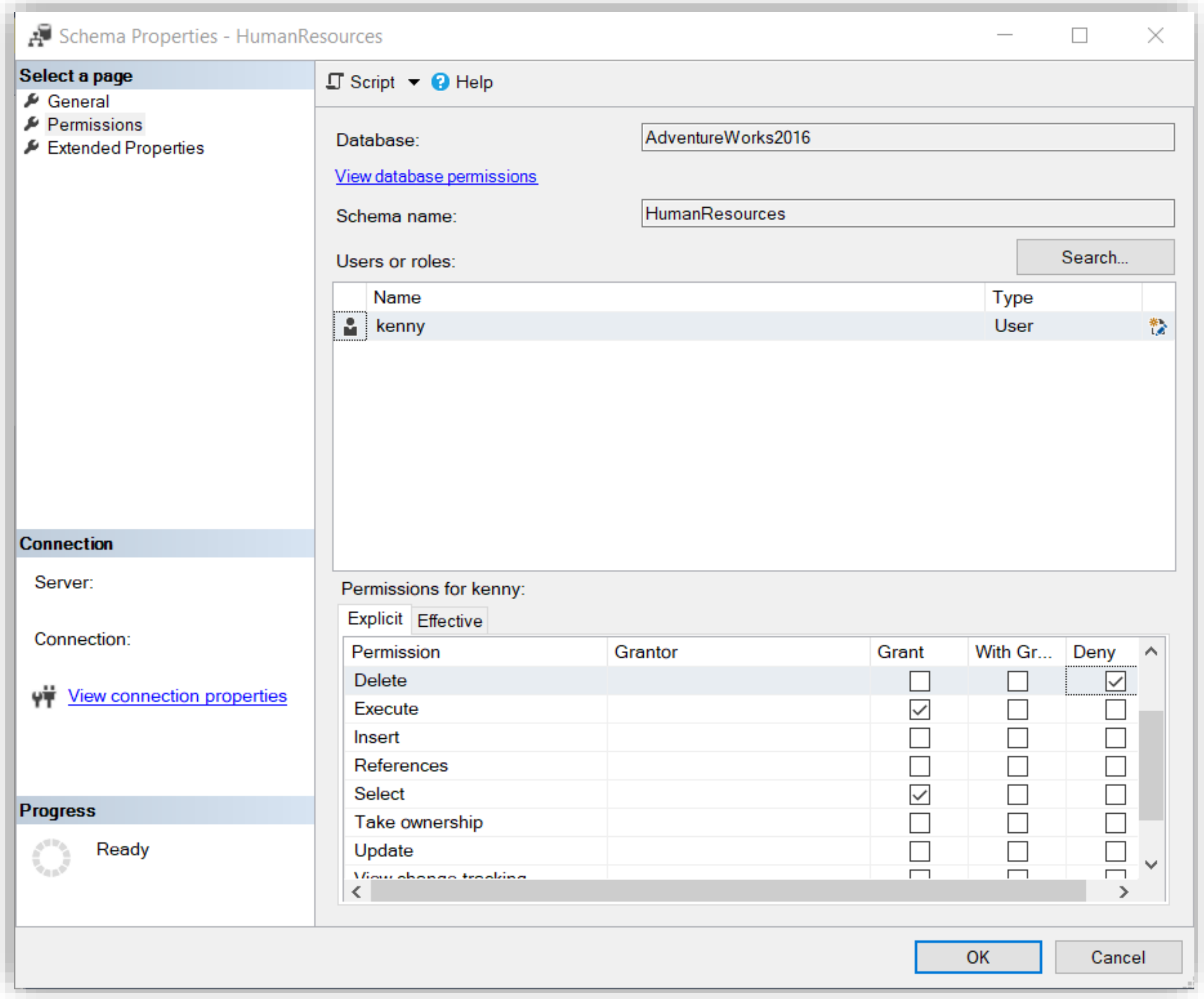
Assign Permissions to a Schema

Permissions assigned at the schema level affect all objects belonging to that schema.

Tables and Views can be assigned the same permissions.

The Execute permission will be applied to all Stored Procedures in the schema.

Select the Effective Tab to see what permissions have been granted.



Database: AdventureWorks2016

View database permissions

Schema name: HumanResources

Users or roles: Search...

Name	Type
kenny	User

Permissions for kenny:

Explicit | Effective

Permission	Grantor	Grant	With Gr...	Deny
Delete		☐	☐	✓
Execute		✓	☐	☐
Insert		☐	☐	☐
References		☐	☐	☐
Select		✓	☐	☐
Take ownership		☐	☐	☐
Update		☐	☐	☐

Questions?

Knowledge Check

What is the difference between Authentication and Authorization?

What are the two types of Logins for SQL Server?

What is an example of a securable?

What are the two Authentication modes?

How should the (sa) account been handled?

How would an admin explicitly restrict access to a table?

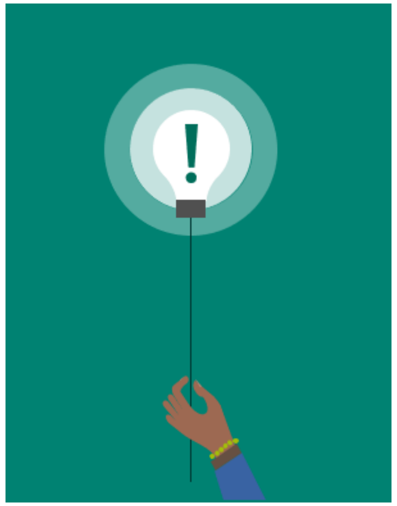
What statement would be used to remove a permission?

Lesson 2: Row Level Security

Objectives

After completing this learning, you will be able to:

- Understand row-level security and how it can be used.



Row-Level Security Overview

Enables fine grained access control at the row level

Security logic is controlled at the database tier instead of the application tier

Row Level Security Scenarios

A hospital can restrict doctors and nurses to only view data about their specific patients.

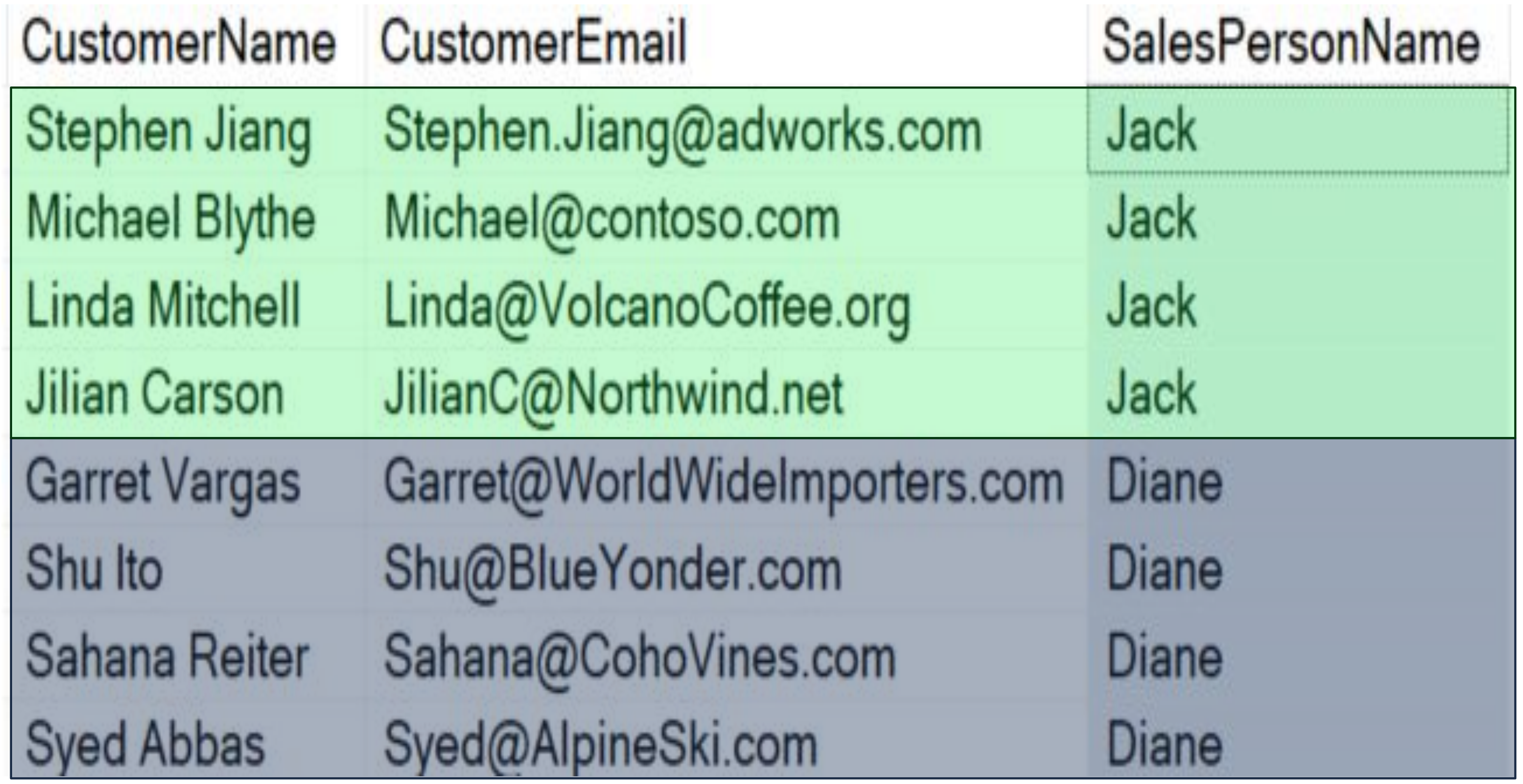
A bank can restrict access to data based on the location of their branch offices.

A bicycle company can restrict sales leads to only specific salespeople.

Salespeople for the Adventure Works Bicycle Company

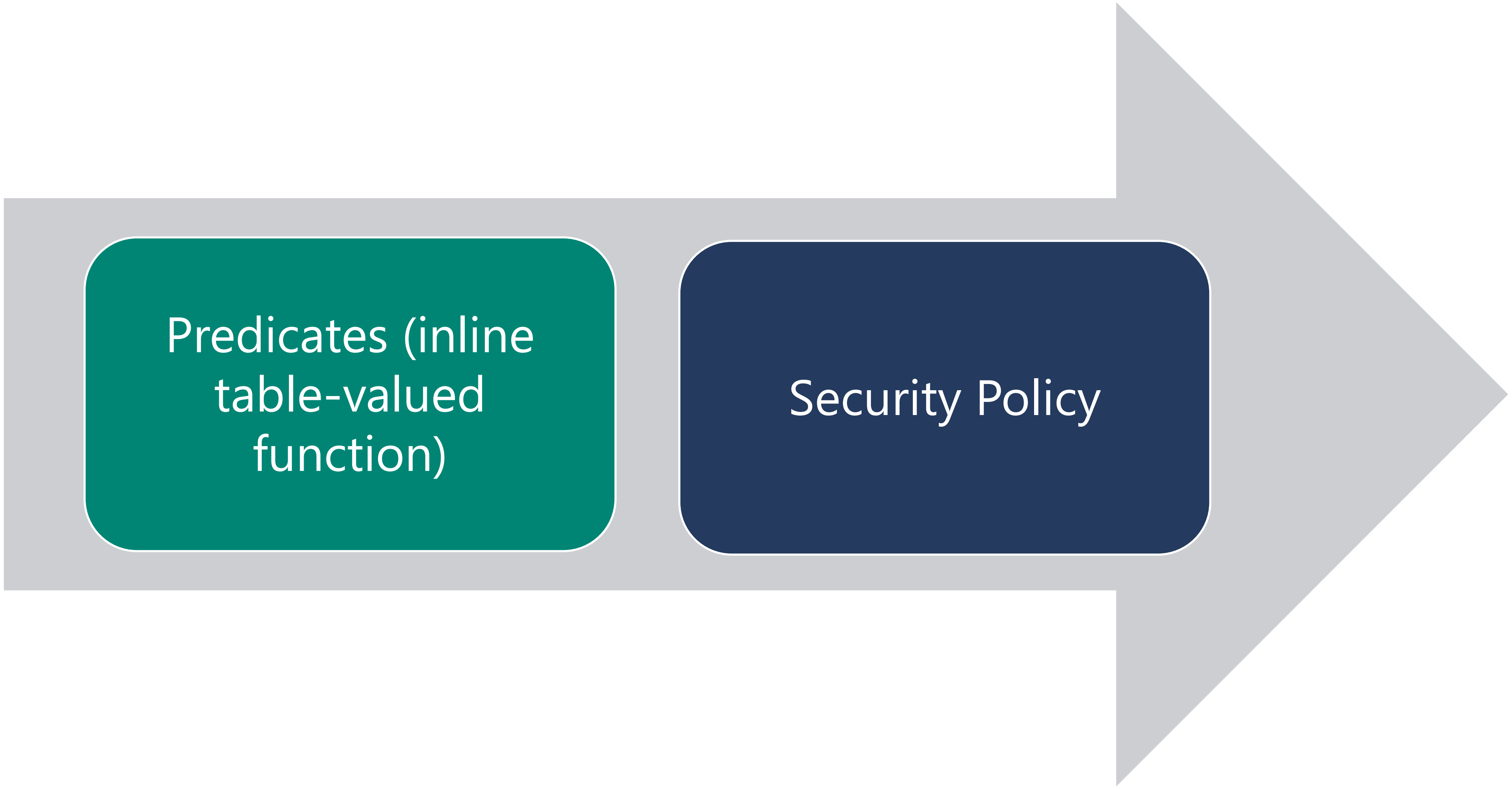
Jack

Diane



CustomerName	CustomerEmail	SalesPersonName
Stephen Jiang	Stephen.Jiang@adworks.com	Jack
Michael Blythe	Michael@contoso.com	Jack
Linda Mitchell	Linda@VolcanoCoffee.org	Jack
Jilian Carson	JilianC@Northwind.net	Jack
Garret Vargas	Garret@WorldWideImporters.com	Diane
Shu Ito	Shu@BlueYonder.com	Diane
Sahana Reiter	Sahana@CohoVines.com	Diane
Syed Abbas	Syed@AlpineSki.com	Diane

Row-Level Security Components



Row-Level Security Predicates

Filter

Block

```
CREATE FUNCTION fn_RowLevelSecurity (@FilterColumnName sysname) RETURNS TABLE WITH
SCHEMABINDING
as
RETURN SELECT 1 as fn_SecureCustomerData
-- filter out records based on database user name
where @FilterColumnName = user_name();
```

Row-Level Security Policy

Security policies are named objects, scoped to a schema that perform filtering using an inline table-valued function.

State setting determines if they are on or off.

```
CREATE SECURITY POLICY FilterCustomer
ADD FILTER PREDICATE dbo.fn_RowLevelSecurity(SalesPersonUserName)
ON dbo.Customer
WITH (STATE = ON);
```

Row-Level Security Permissions

Create, alter or drop policy

- Requires ALTER ANY SECURITY POLICY
- Creating or dropping a security policy requires the ALTER permission on the schema

For each predicate added

- SELECT and REFERENCES permissions on the function being used as a predicate
- REFERENCES on the target table
- REFERENCES on every column from the target table used as an argument

Row-Level Security Best Practices

Create a separate schema for RLS objects

Monitor who has the ALTER ANY SECURITY POLICY – intended for highly privileged users

If the security policy managers have the ALTER ANY SECURITY POLICY permission, they do not need the select permission on the table

Keep predicate functions as simple as possible to prevent performance issues

Row-Level Security Limitations

- Filestream – not supported
- Polybase – not supported
- DBCC SHOW_STATISTICS report statistics on unfiltered data (potential leak)
- Memory-optimized tables – predicate must use WITH NATIVE_COMPILATION option
- Indexed views cannot be created on top of tables that have a security policy
- Change Data Capture (CDC) – can leak rows that should be filtered to db_owner

Demonstration

Row-Level Security

Create a RLS Function and Security Policy

Row-Level Security

You will setup row level security to allow different people to see their territory without seeing data for other territories.

Questions?

Knowledge Check

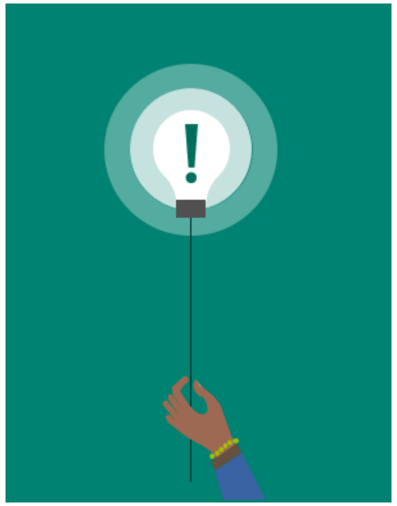
What are some scenarios where row-level security would be beneficial?

Lesson 3: Dynamic Data Masking

Objectives

After completing this learning, you will be able to:

- Understand how dynamic data masking can be utilized to enhance data security.



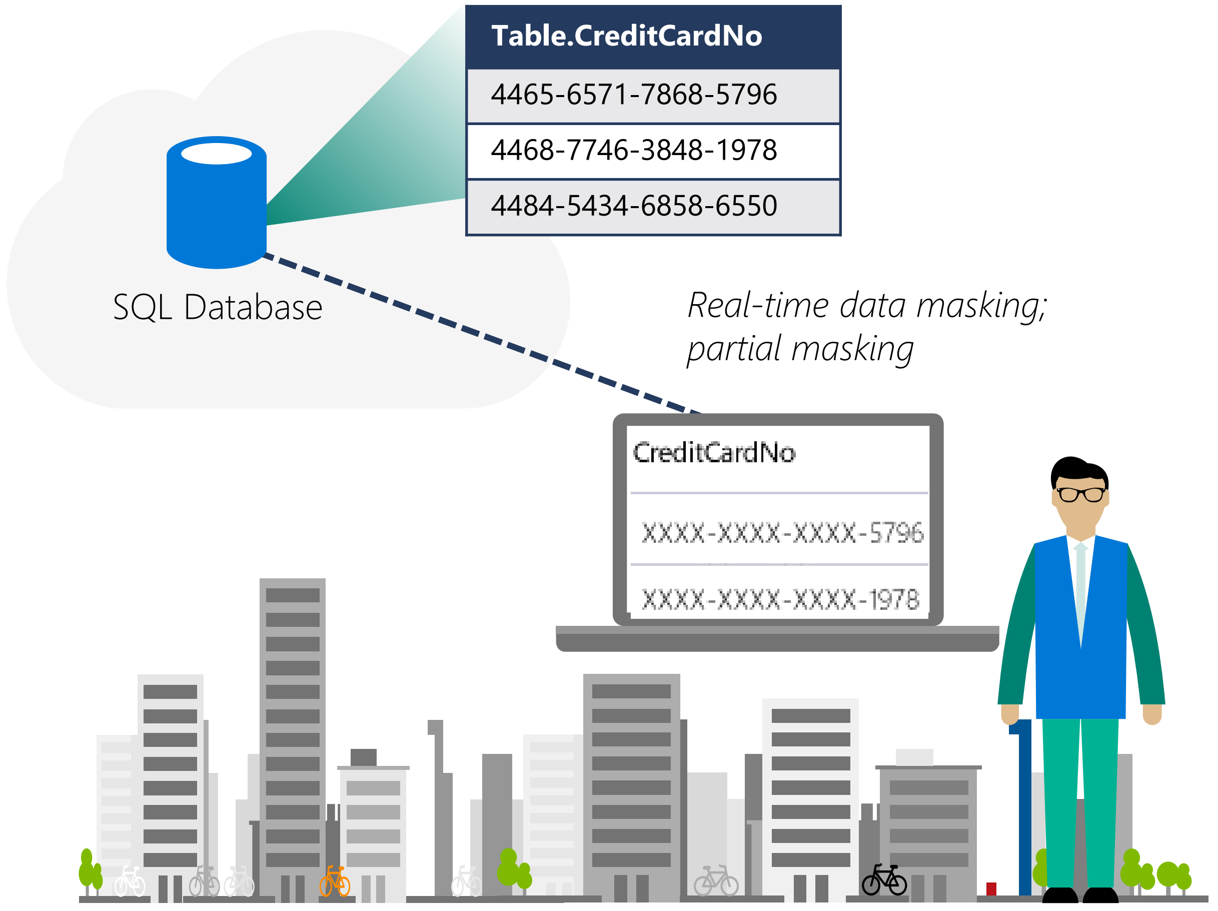
Dynamic Data Masking

Prevent abuse of sensitive data by hiding it from users

Policy-driven at table and column level for defined set of users

Applied in real time to query results based on policy

Multiple masking functions available for various sensitive data categories



Dynamic Data Masking scenarios

Developers can troubleshoot production data without viewing sensitive information.

Customer Service representatives can view parts of sensitive data like credit card information.

Reports can be distributed with sensitive data obfuscated at the data layer.

Types of Data Masks

Default (based on data type)

Email

Custom String

Random

Dynamic Data Masking Limitations

The following columns cannot be masked

- Using Always Encrypted
- FILESTREAM
- Computed Columns*
- Column that is a key for a full-text index

Dynamic Data Masking Permissions

- ALTER ANY MASK and ALTER permission on the table
  - Required to add, replace or remove the mask of a column
- UNMASK
  - Required to see unmasked data*

Demonstration

Dynamic Data Masking

Creating and Querying Masked Tables

Questions?

Knowledge Check

Which permission needs to be granted for a user to see the full data view

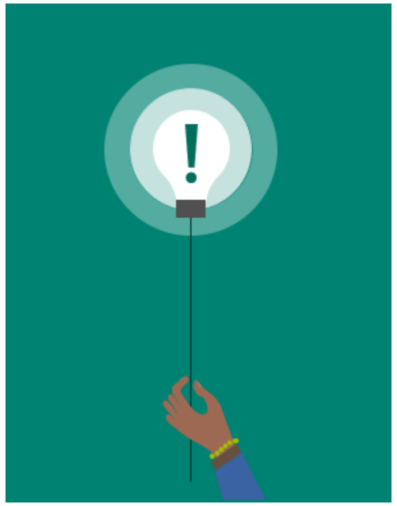
What are the four types of data masks?

Lesson 4: Introduction to SQL Server Audit

Objectives

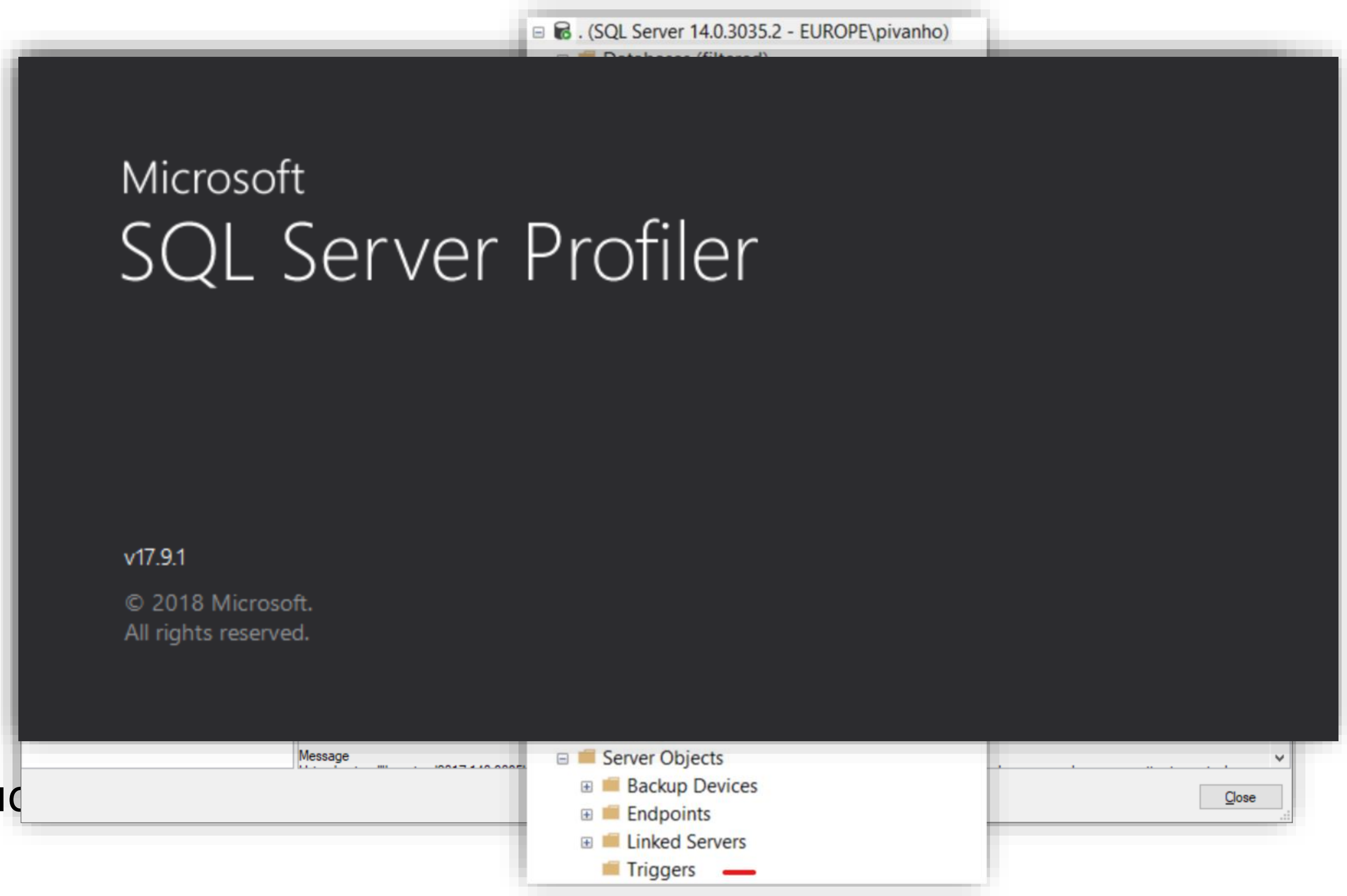
After completing this learning, you will be able to:

- Understand what SQL Server Audit is and how to configure it.



Some History...

- SQL 2005 and earlier
  - Server Level
  - Application Log
  - SQL Server Error Log
- Triggers
  - Login triggers
  - Server triggers
  - DDL triggers
- SQL Trace (Profiler)
  - Detailed activity audits
  - Individual statements, incluc



SQL Server Audit

Provides the ability to track and log events that occur in SQL Server engine

Built on Extended Events architecture

Basic Server Audit available in all SKUs (Since SQL Server 2012)

- Database Audit only available in Enterprise, Developer, and Evaluation editions

SQL Server Audit Components

- Allow pre-filtering and fine-grained auditing
- Allow multiple targets (File, Application and Security Event Logs)

Key part of security strategy

Who has accessed or attempted to access your data

Ability to detect unauthorized access attempts

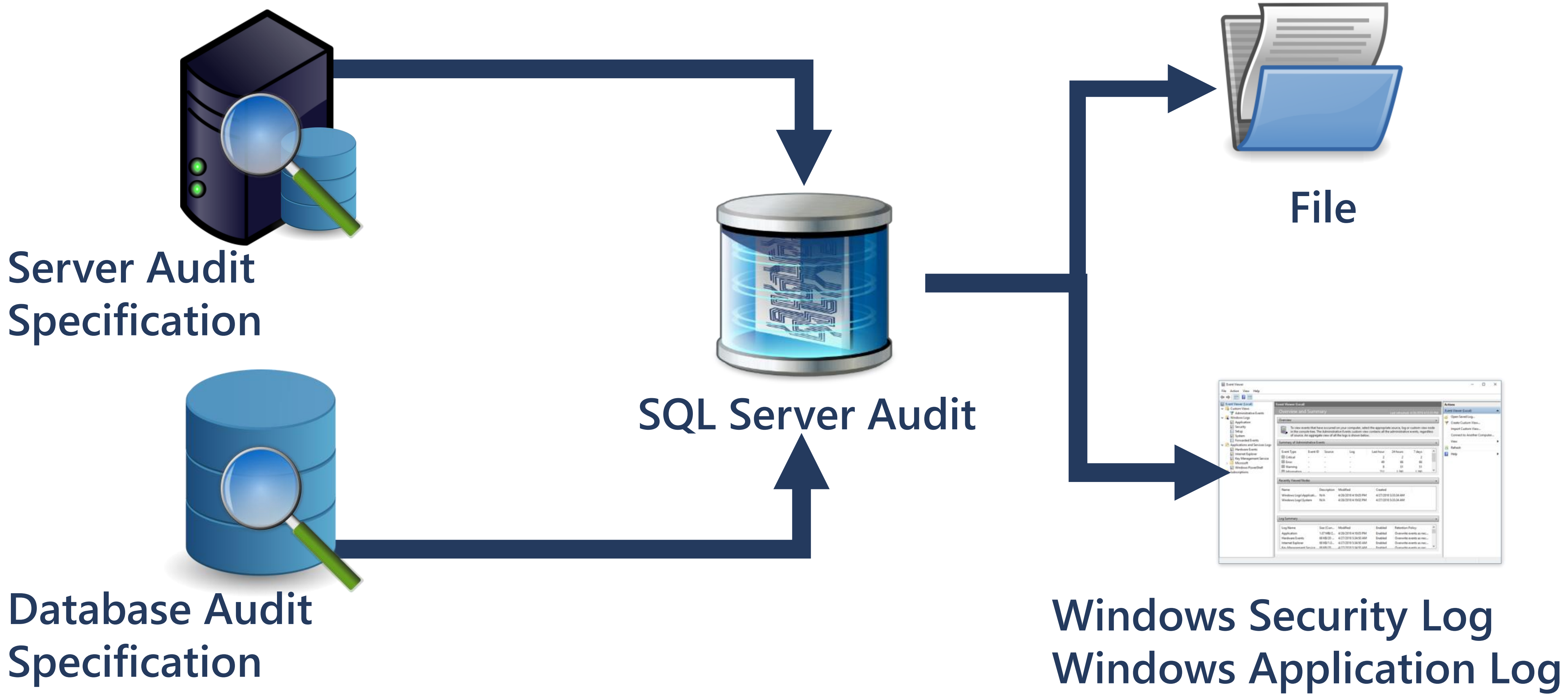
Piece together the actions of malicious insiders

Robust tracking capability

Primary Goals of SQL Server Audit

Security	• The audit feature must be truly secure.
Performance	• Performance impact must be minimized
Management	• The audit feature must be easy to manage.
Discoverability	• Audit-centric questions must be easy to answer

Audit Object Layout



Working with SQL Server Audit

- Create an audit and define the target
- Create either a server audit specification or database audit specification
- Enable the audit specification
- Enable the audit
- Read the audit events

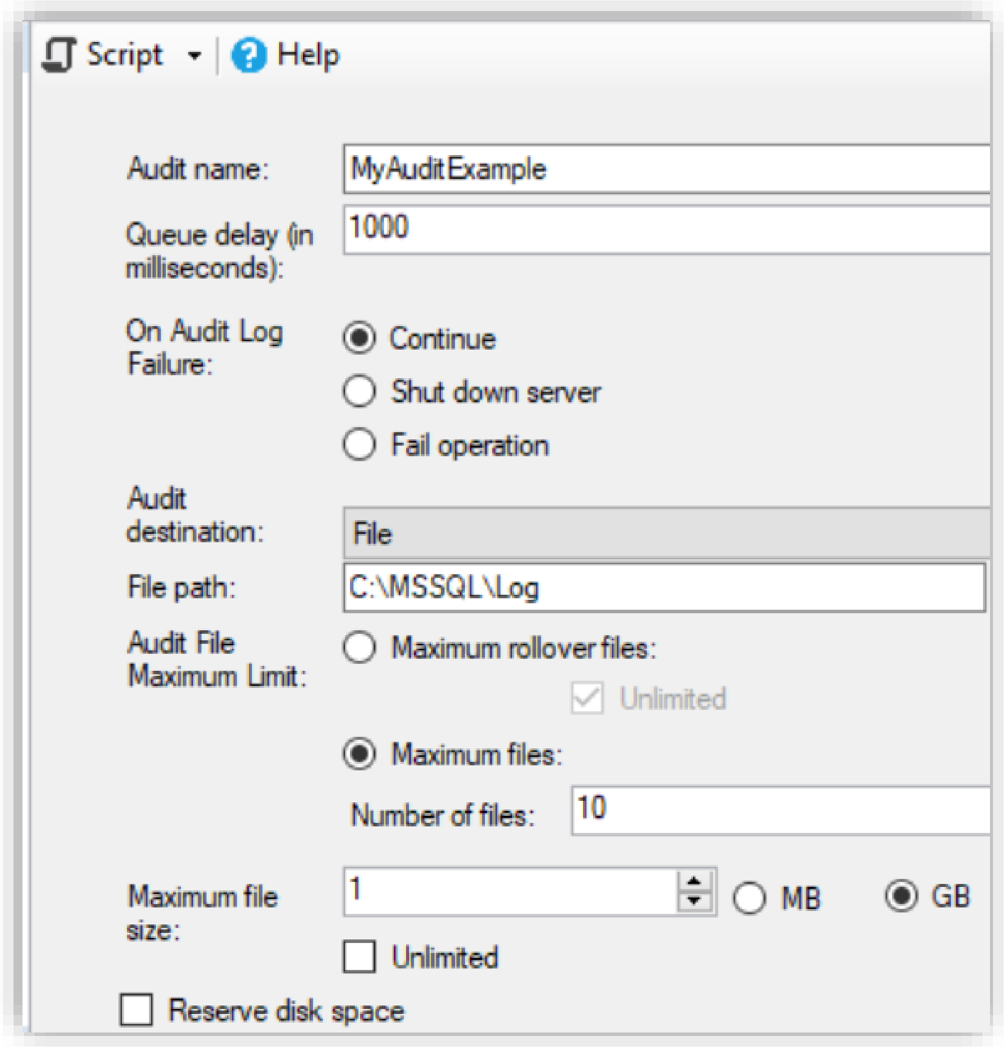
Create Audit

Queue delay (in milliseconds)

On Audit Log Failure - Continue

On Audit Log Failure - Shut down server

On Audit Log Failure - Fail operation



Audit name:	MyAuditExample
Queue delay (in milliseconds):	1000
On Audit Log Failure:	[✓ Continue] Shut down server Fail operation
Audit destination:	File
File path:	C:\MSSQL\Log
Audit File Maximum Limit:	Maximum rollover files: ✓ Unlimited [✓ Maximum files:]
Number of files:	10
Maximum file size:	1 MB [✓ GB] ☐ Unlimited

☐ Reserve disk space

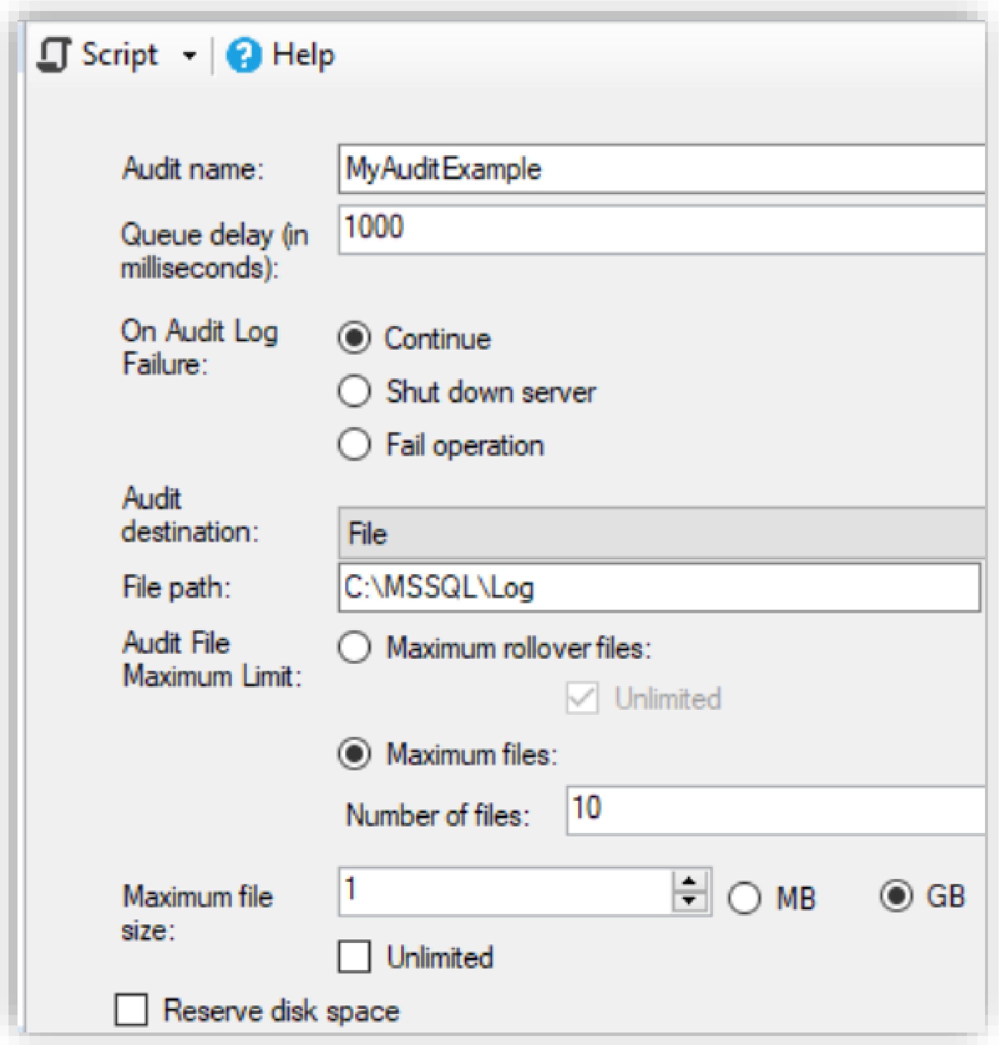
Create Audit (Continued)

Audit Destination

- Binary file
- Windows Application log
- Windows Security log

File Settings

- File Path
- Audit File Maximum Limit
- Maximum File Size
- Reserve disk space



Audit name: MyAuditExample

Queue delay (in milliseconds): 1000

On Audit Log Failure: [✓ Continue] Shut down server Fail operation

Audit destination: File

File path: C:\MSSQL\Log

Audit File Maximum Limit: Maximum rollover files: [✓ Unlimited] [✓ Maximum files:]

Number of files: 10

Maximum file size: 1 MB [✓ GB]

☐ Unlimited

☐ Reserve disk space

Create an Audit Specification (Syntax)

Database Audit Specification

```
CREATE DATABASE AUDIT SPECIFICATION
audit_specification_name
{ FOR SERVER AUDIT audit_name
[
{ ADD ( { <audit_action_specification> |
audit_action_group_name } ) }
[, ...n] ]
[ WITH ( STATE = { ON | OFF } ) ] }
[ ; ] <audit_action_specification>::= { action [
,...n ]ON [ class :: ] securable BY principal [ ,...n
] }
```

Server Audit Specification

```
CREATE SERVER AUDIT SPECIFICATION
audit_specification_name
FOR SERVER AUDIT audit_name
{
{ ADD ( { audit_action_group_name } ) }
[, ...n]
[ WITH ( STATE = { ON | OFF } ) ] } [ ; ]
```

SQL Server Audit Events to the Security Log

The Audit object

- The Audit object access setting must be configured to capture the events. The audit policy tool (auditpol.exe) exposes a variety of sub-policies settings in the audit object access category. To allow SQL Server to audit object access, configure the application generated setting.

SQL Server service Account

- The account that the SQL Server service is running under must have the generate security audits permission to write to the Windows Security log.
- secpol.msc → Generate security audits

Registry

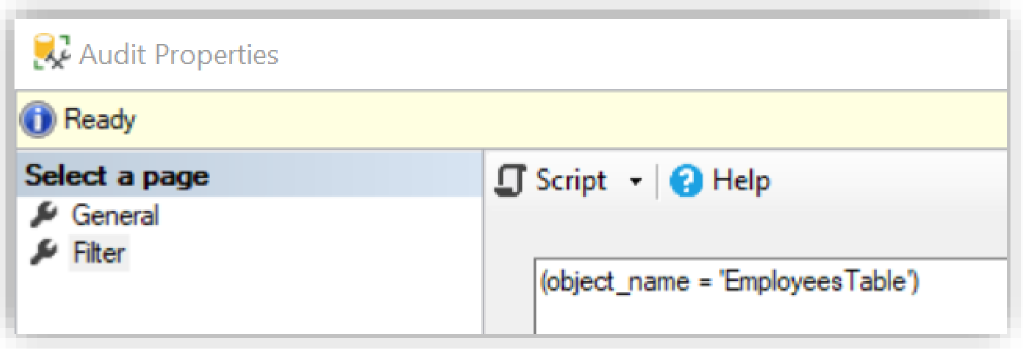
- Provide full permission for the SQL Server service account to the registry hive
- HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog\Security.

Create Audit Filter

Enter a predicate, or “WHERE clause”

Audit events are filtered before they are written to the audit log

You can filter on every element of the Audit Records



```
(object_name = 'EmployeesTable')
```

Server-Level Audit Action Groups

LOGIN_CHANGE_PASSWORD_GROUP

- Whenever a login password is changed

SERVER_OBJECT_CHANGE_GROUP

- CREATE, ALTER, or DROP operations on server objects

SERVER_PRINCIPAL_CHANGE_GROUP

- When server principals are created, altered, or dropped

SERVER_ROLE_MEMBER_CHANGE_GROUP

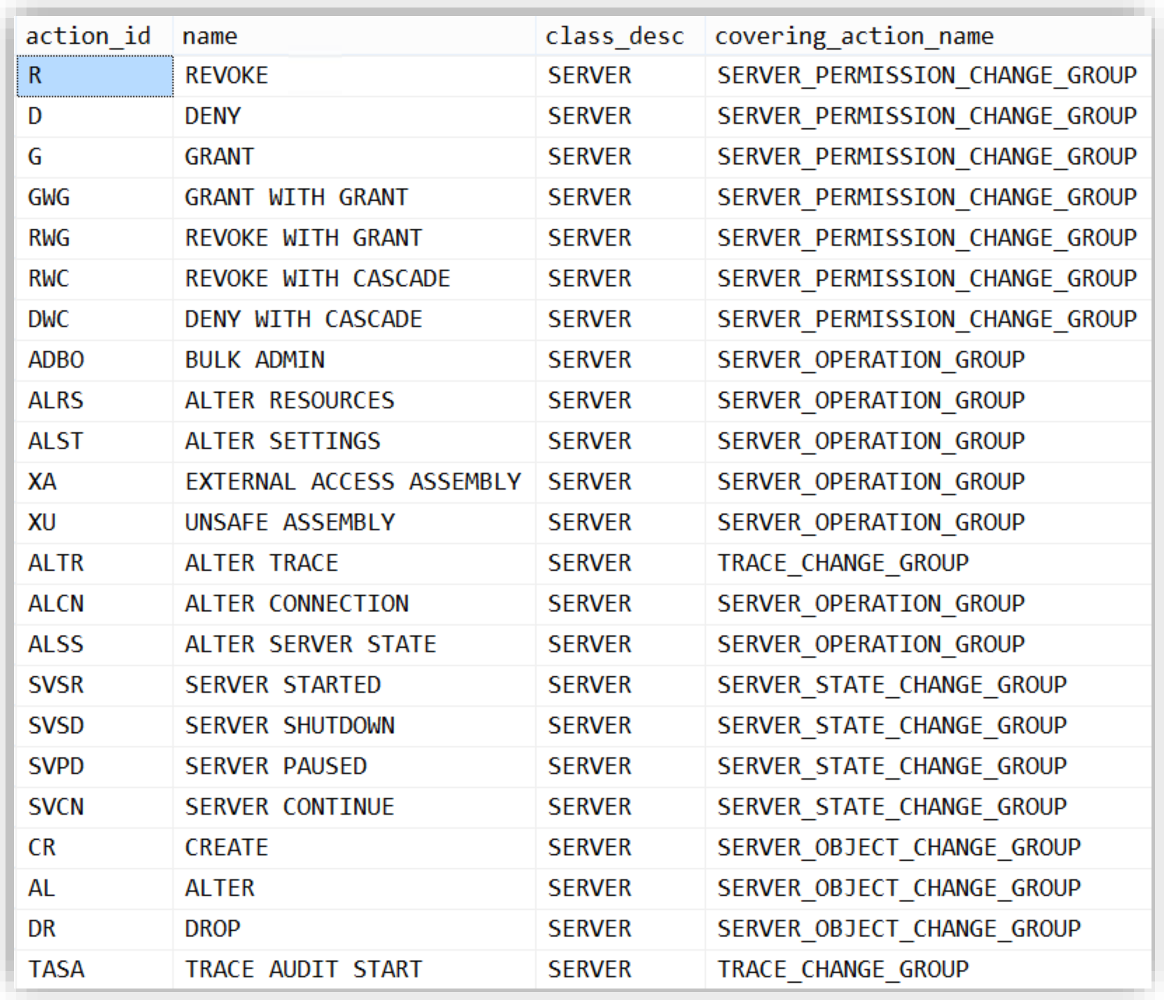
- Whenever a login is added or removed from a fixed server role.

SUCCESSFUL_LOGIN_GROUP

- A principal has successfully logged in to SQL Server

Server Audit Specifications Actions and Groups

```sql
SELECT *
FROM sys.dm_audit_actions
WHERE class_desc = 'server';
```



action_id	name	class_desc	covering_action_name
R	REVOKE	SERVER	SERVER_PERMISSION_CHANGE_GROUP
D	DENY	SERVER	SERVER_PERMISSION_CHANGE_GROUP
G	GRANT	SERVER	SERVER_PERMISSION_CHANGE_GROUP
GWG	GRANT WITH GRANT	SERVER	SERVER_PERMISSION_CHANGE_GROUP
RWG	REVOKE WITH GRANT	SERVER	SERVER_PERMISSION_CHANGE_GROUP
RWC	REVOKE WITH CASCADE	SERVER	SERVER_PERMISSION_CHANGE_GROUP
DWC	DENY WITH CASCADE	SERVER	SERVER_PERMISSION_CHANGE_GROUP
ADBO	BULK ADMIN	SERVER	SERVER_OPERATION_GROUP
ALRS	ALTER RESOURCES	SERVER	SERVER_OPERATION_GROUP
ALST	ALTER SETTINGS	SERVER	SERVER_OPERATION_GROUP
XA	EXTERNAL ACCESS ASSEMBLY	SERVER	SERVER_OPERATION_GROUP
XU	UNSAFE ASSEMBLY	SERVER	SERVER_OPERATION_GROUP
ALTR	ALTER TRACE	SERVER	TRACE_CHANGE_GROUP
ALCN	ALTER CONNECTION	SERVER	SERVER_OPERATION_GROUP
ALSS	ALTER SERVER STATE	SERVER	SERVER_OPERATION_GROUP
SVSR	SERVER STARTED	SERVER	SERVER_STATE_CHANGE_GROUP
SVSD	SERVER SHUTDOWN	SERVER	SERVER_STATE_CHANGE_GROUP
SVPD	SERVER PAUSED	SERVER	SERVER_STATE_CHANGE_GROUP
SVCN	SERVER CONTINUE	SERVER	SERVER_STATE_CHANGE_GROUP
CR	CREATE	SERVER	SERVER_OBJECT_CHANGE_GROUP
AL	ALTER	SERVER	SERVER_OBJECT_CHANGE_GROUP
DR	DROP	SERVER	SERVER_OBJECT_CHANGE_GROUP
TASA	TRACE AUDIT START	SERVER	TRACE_CHANGE_GROUP

Database-Level Audit Action Groups

BACKUP_RESTORE_GROUP

- Whenever a backup or restore command is issued

DATABASE_CHANGE_GROUP

- When a database is created, altered, or dropped

DATABASE_OBJECT_CHANGE_GROUP

- When a CREATE, ALTER, or DROP statement is executed on database objects

DATABASE_ROLE_MEMBER_CHANGE_GROUP

- Whenever a login is added to or removed from a database role

DBCC_GROUP

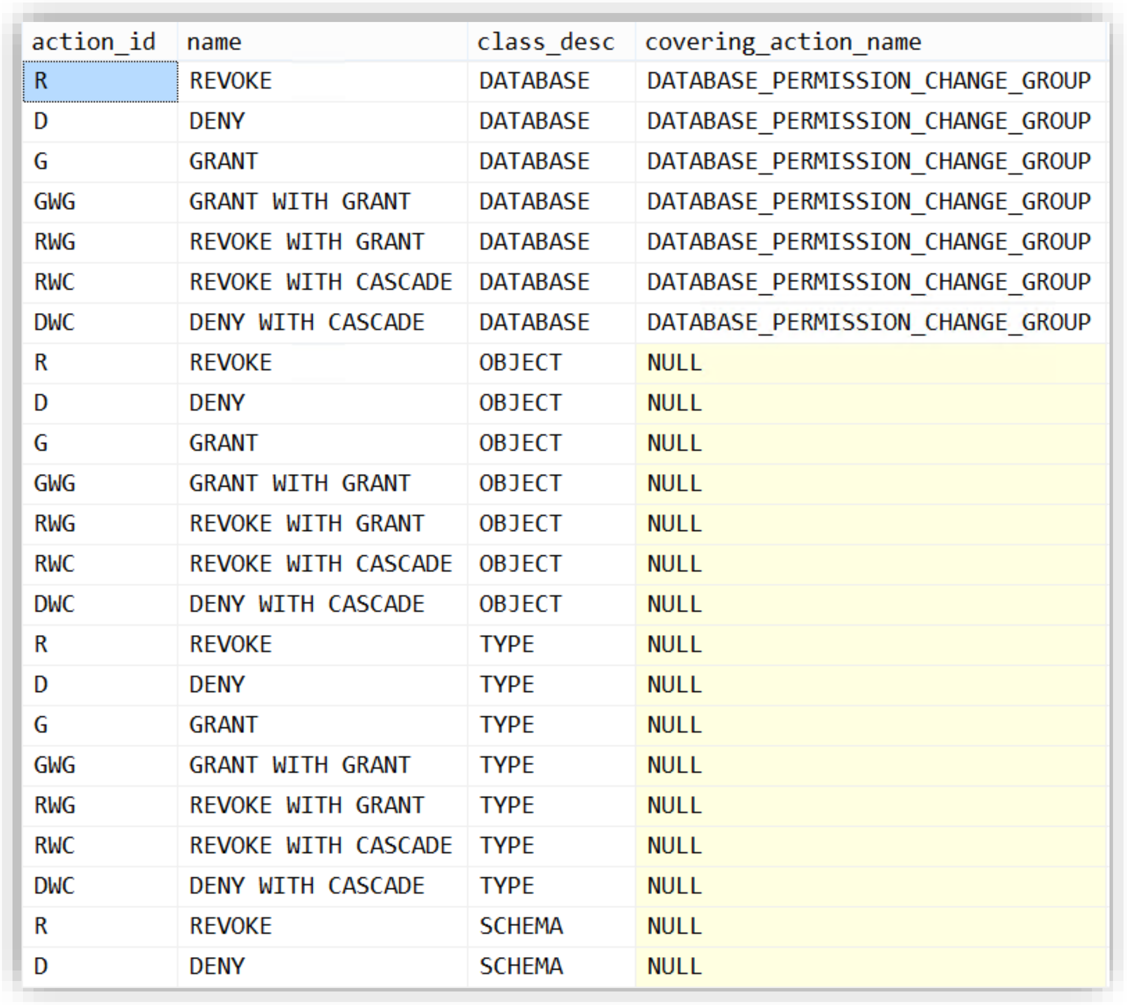
- Whenever a principal issues any DBCC command

FAILED_DATABASE_AUTHENTICATION_GROUP

- A principal tried to log on to SQL Server and failed

Database Audit Specifications Actions and Groups

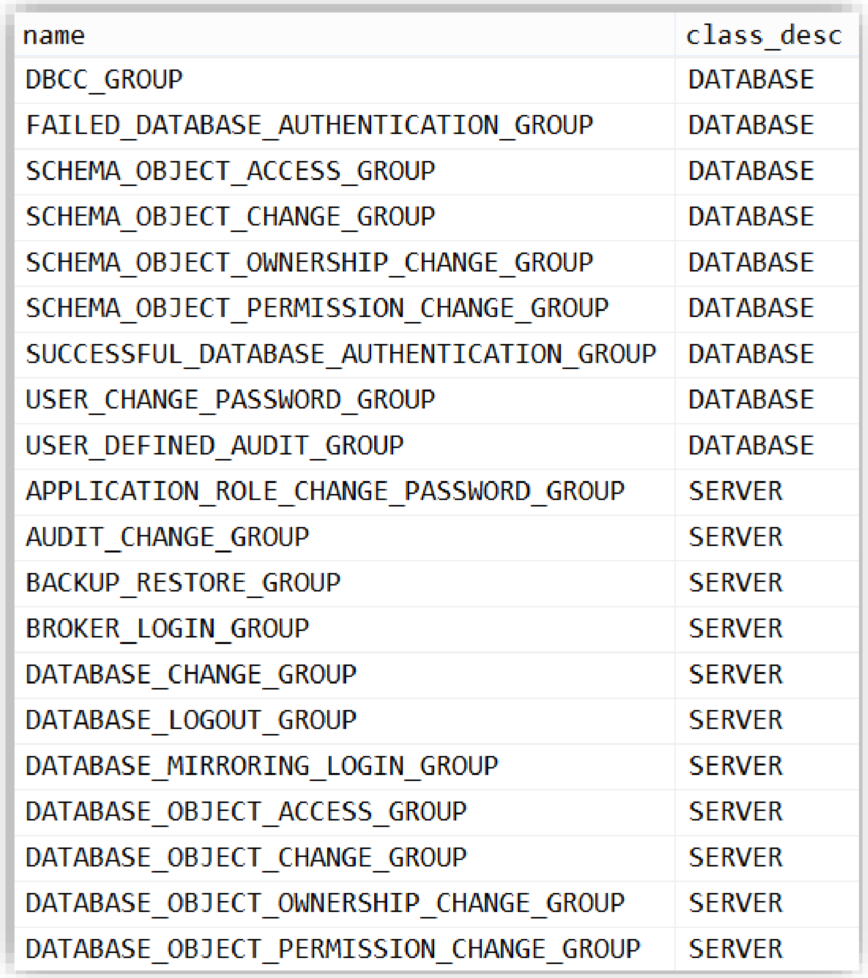
```sql
SELECT *
FROM sys.dm_audit_actions
WHERE class_desc = 'database'
OR parent_class_desc = 'database';
```



action_id	name	class_desc	covering_action_name
R	REVOKE	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
D	DENY	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
G	GRANT	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
GWG	GRANT WITH GRANT	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
RWG	REVOKE WITH GRANT	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
RWC	REVOKE WITH CASCADE	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
DWC	DENY WITH CASCADE	DATABASE	DATABASE_PERMISSION_CHANGE_GROUP
R	REVOKE	OBJECT	NULL
D	DENY	OBJECT	NULL
G	GRANT	OBJECT	NULL
GWG	GRANT WITH GRANT	OBJECT	NULL
RWG	REVOKE WITH GRANT	OBJECT	NULL
RWC	REVOKE WITH CASCADE	OBJECT	NULL
DWC	DENY WITH CASCADE	OBJECT	NULL
R	REVOKE	TYPE	NULL
D	DENY	TYPE	NULL
G	GRANT	TYPE	NULL
GWG	GRANT WITH GRANT	TYPE	NULL
RWG	REVOKE WITH GRANT	TYPE	NULL
RWC	REVOKE WITH CASCADE	TYPE	NULL
DWC	DENY WITH CASCADE	TYPE	NULL
R	REVOKE	SCHEMA	NULL
D	DENY	SCHEMA	NULL

List All Server and Database Action Groups

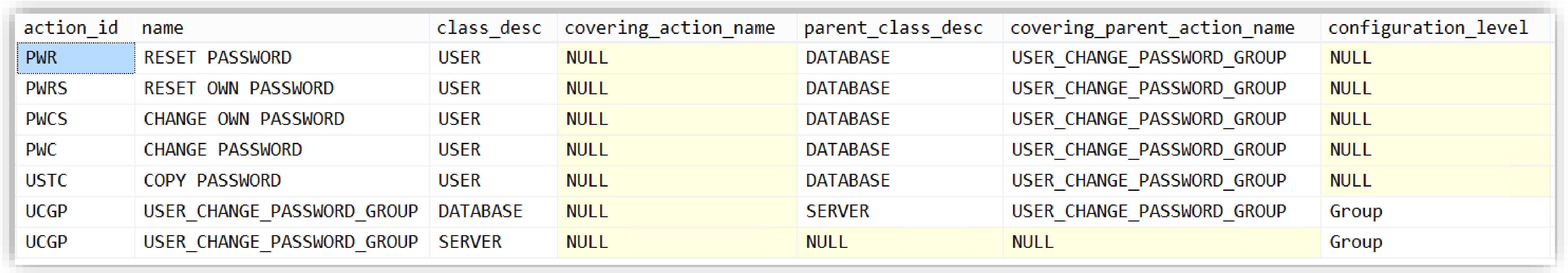
```
SELECT name, class_desc
FROM sys.dm_audit_actions
WHERE name IN
    (SELECT containing_group_name
    FROM sys.dm_audit_actions)
ORDER BY class_desc, name;
```



name	class_desc
DBCC_GROUP	DATABASE
FAILED_DATABASE_AUTHENTICATION_GROUP	DATABASE
SCHEMA_OBJECT_ACCESS_GROUP	DATABASE
SCHEMA_OBJECT_CHANGE_GROUP	DATABASE
SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP	DATABASE
SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP	DATABASE
SUCCESSFUL_DATABASE_AUTHENTICATION_GROUP	DATABASE
USER_CHANGE_PASSWORD_GROUP	DATABASE
USER_DEFINED_AUDIT_GROUP	DATABASE
APPLICATION_ROLE_CHANGE_PASSWORD_GROUP	SERVER
AUDIT_CHANGE_GROUP	SERVER
BACKUP_RESTORE_GROUP	SERVER
BROKER_LOGIN_GROUP	SERVER
DATABASE_CHANGE_GROUP	SERVER
DATABASE_LOGOUT_GROUP	SERVER
DATABASE_MIRRORING_LOGIN_GROUP	SERVER
DATABASE_OBJECT_ACCESS_GROUP	SERVER
DATABASE_OBJECT_CHANGE_GROUP	SERVER
DATABASE_OBJECT_OWNERSHIP_CHANGE_GROUP	SERVER
DATABASE_OBJECT_PERMISSION_CHANGE_GROUP	SERVER

Get Information About a Particular Group Name

```sql
SELECT *
FROM sys.dm_audit_actions
WHERE containing_group_name = 'USER_CHANGE_PASSWORD_GROUP';
```



action_id	name	class_desc	covering_action_name	parent_class_desc	covering_parent_action_name	configuration_level
PWR	RESET PASSWORD	USER	NULL	DATABASE	USER_CHANGE_PASSWORD_GROUP	NULL
PWRS	RESET OWN PASSWORD	USER	NULL	DATABASE	USER_CHANGE_PASSWORD_GROUP	NULL
PWCS	CHANGE OWN PASSWORD	USER	NULL	DATABASE	USER_CHANGE_PASSWORD_GROUP	NULL
PWC	CHANGE PASSWORD	USER	NULL	DATABASE	USER_CHANGE_PASSWORD_GROUP	NULL
USTC	COPY PASSWORD	USER	NULL	DATABASE	USER_CHANGE_PASSWORD_GROUP	NULL
UCGP	USER_CHANGE_PASSWORD_GROUP	DATABASE	NULL	SERVER	USER_CHANGE_PASSWORD_GROUP	Group
UCGP	USER_CHANGE_PASSWORD_GROUP	SERVER	NULL	NULL	NULL	Group

Database-Level Audit Actions

- SELECT
- UPDATE
- INSERT
- DELETE
- EXECUTE
- RECEIVE
- REFERENCES

View a SQL Server Audit Log

SQL SERVER
MANAGEMENT STUDIO

SYS.FN_GET_AUDIT_FILE

sys.fn_get_audit_file

- **file_pattern**
  - Specifies the directory or path and file name for the audit file set to be read.
- **initial_file_name**
  - Specifies the path and name of a specific file in the audit file set to start reading audit records from
- **audit_record_offset**
  - Specifies a known location with the file specified for the initial_file_name

```
SELECT * FROM sys.fn_get_audit_file
('\\serverName\Audit\HIPAA_AUDIT.sqlaudit',default,default);
```

Considerations

In the case of a failure during audit initiation, the server will not start.

Attaching a Database with an Audit Defined

Always On Availability Groups and SQL Server Audit

Auditing Administrators

Demonstration

Demonstrate how to Create an Audit and Audit Specification within SQL Server

Questions?

Knowledge Check

How do you start an audit after creating the server audit specification login?

How do you stop the audit?

What are audit action groups in a server audit specification?

Microsoft