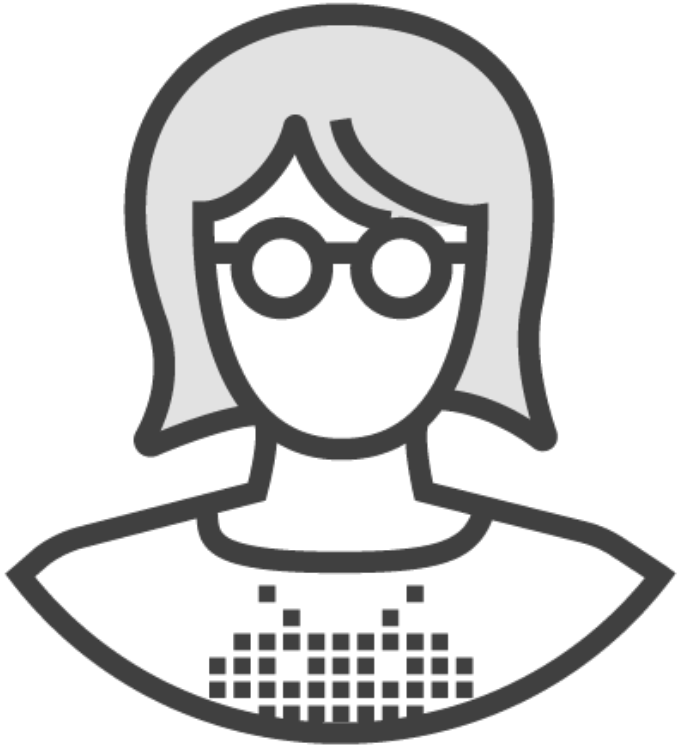
Implementing Security with Users, Groups, and Roles

Russ Thomas
DATABASE MANAGER

@sqljudo www.sqljudo.com

Database Security Challenge

- Every Implementation is Unique
- Security Needs Constantly Evolve
- Options are NOT Few
- Threats to Security Are Everywhere



Your Goal as the Administrator

1. Implement Security that is Manageable

2. Implement Security that is Effective

Module Summary

SQL Server Security

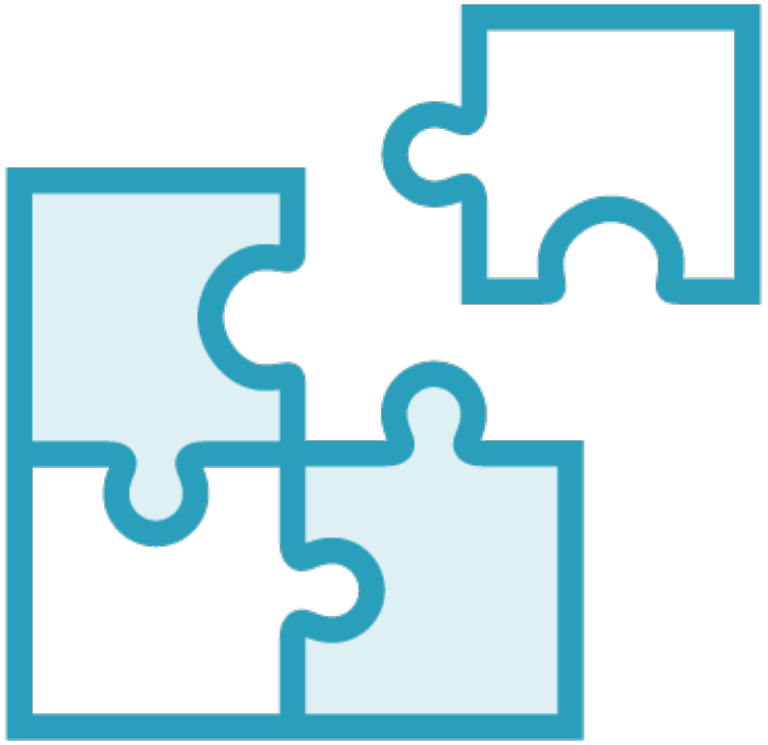
- Principals
- Securables
- Scope

Security Management

- Logins
- Groups
- Roles
- Authentication

Real World Compliance

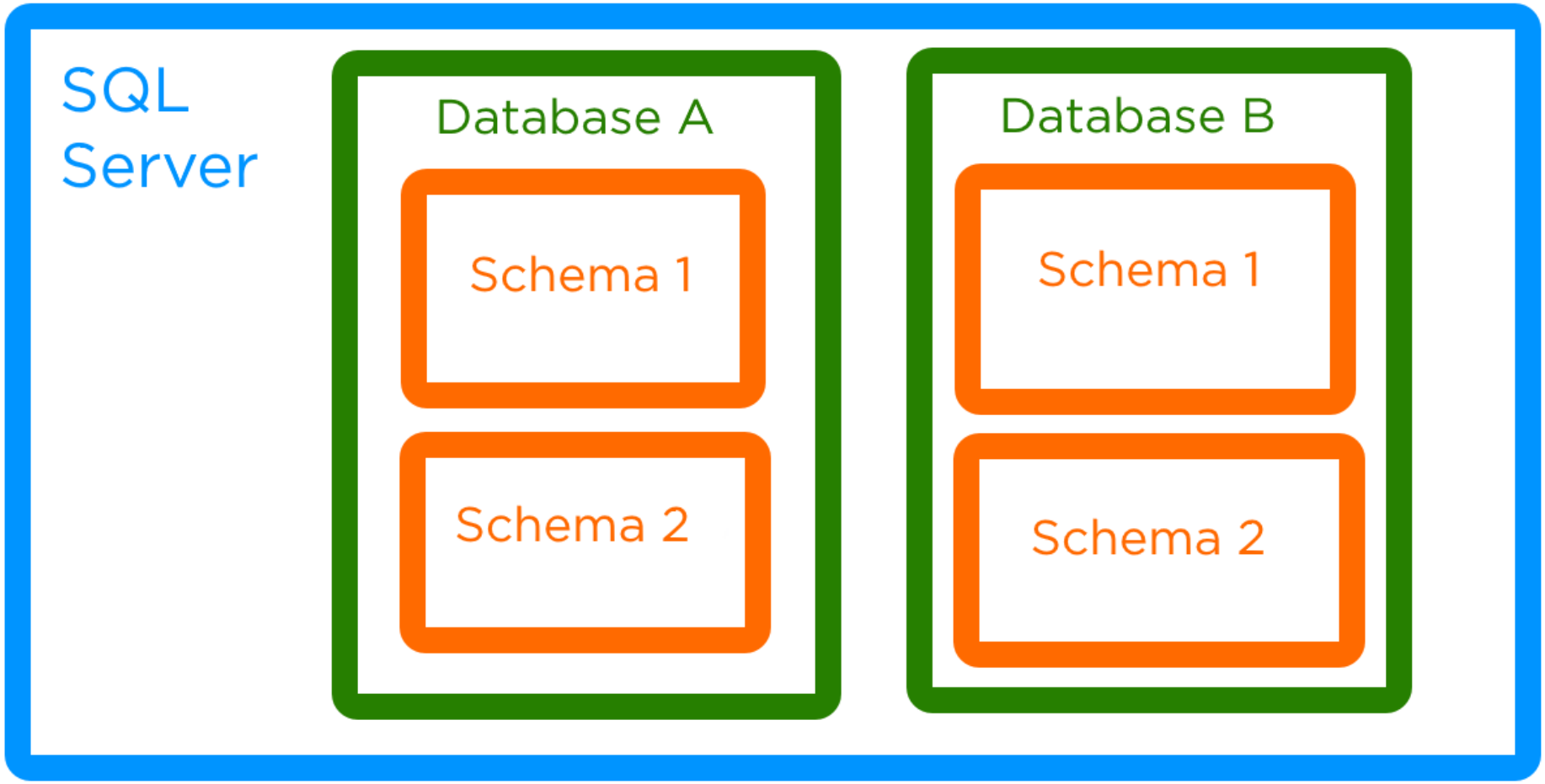
- PCI Example
- HIPAA Example



SCOPE

Scope is the area of effect:

- Server Level
- Database Level
- Schema Level


SQL
Server
Database A
Schema 1
Schema 2
Database B
Schema 1
Schema 2

Principals and Securables

Principals

- Any entity that can request a SQL Server resource

Securables

- The resources that principals can request access or privileges to

Principals

Server

- Domain Login
- Domain Groups
- Local Windows Login
- SQL Server Login

Database

- User

Roles

- Server Level Roles
- Database Level Roles
- Application Roles

Logins and Users

LOGIN

Server Level

USER

Database Level

System Administrator

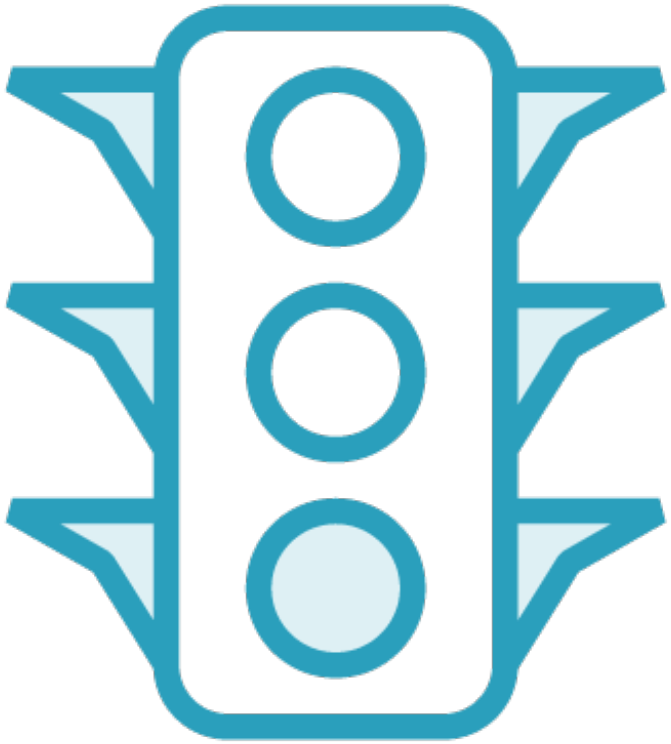
Grants complete privileges across the entire scope of the server.

This goes beyond trust or consideration of intent and ignores basic human nature. We make mistakes.

System Administrator

"rampant random accounts with sys admin privileges on a SQL Server is an indication that at some point your DBA or database engineer just gave up"

Me – I said that



Rule of Least Privilege

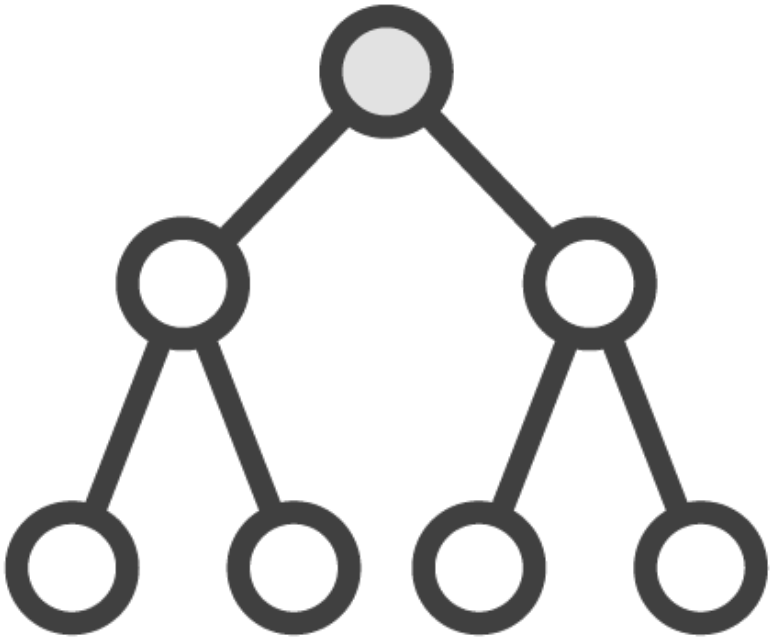
- You only grant access to the minimum level of resources required to accomplish a permitted task

Managing Users and Logins

- create, alter, drop user
- create, alter, drop login

Reference System Tables

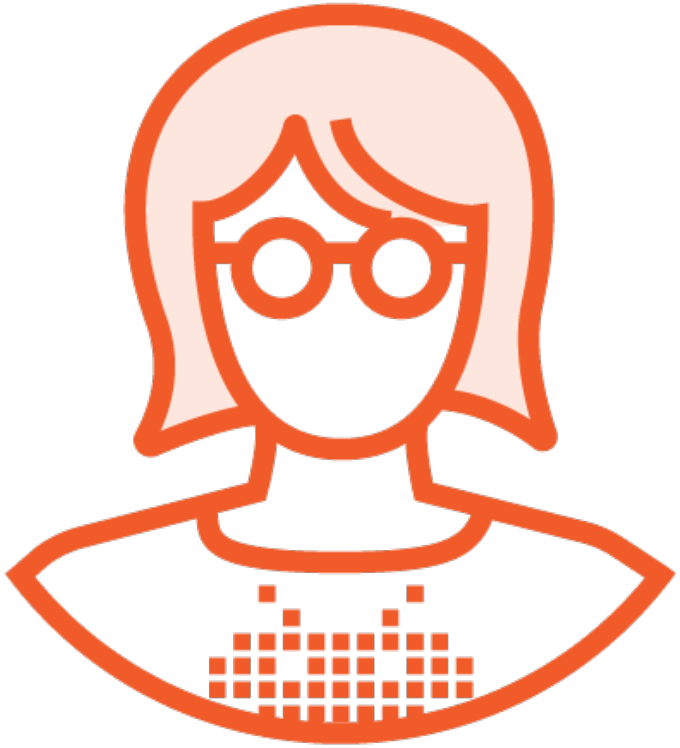
- sys.users
- sys.logins
- sys.database_principals
- sys.database_permissions
- sys.database_role_members



Problems of Scale

Managing security for a single user is typically pretty easy.

- What about many users?
- What about future changes?
- What about ongoing maintenance?



The rule of least effort.

First goal is a robust implementation

Second goal is to do it with as little effort as possible, otherwise we run the risk of failure on both goals.

Up Next:

- Roles

When you leverage domain groups the management of members of that group are taken care of at only one location

Principals

Server

- Domain Login
- Domain Groups
- Local Windows Login
- SQL Server Login

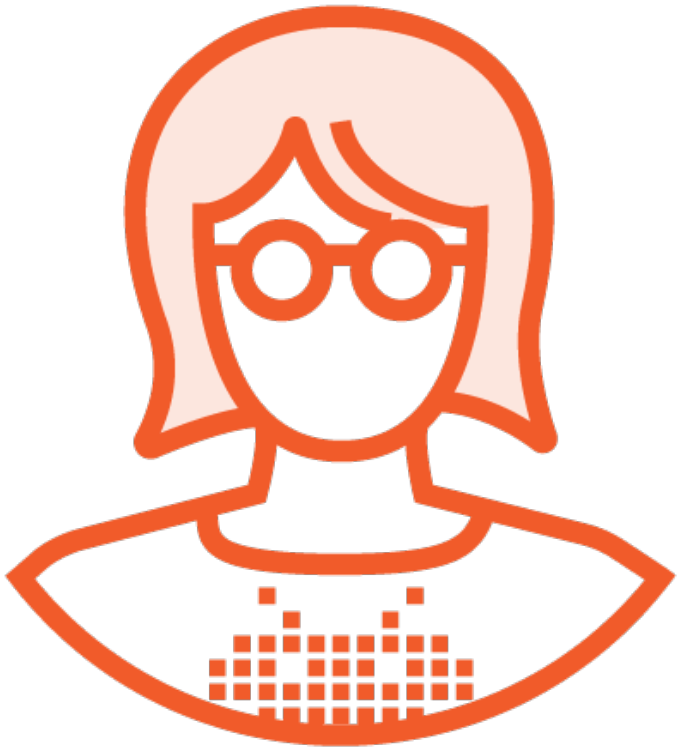
Database

- User

Roles

- Server Level Roles
- Database Level Roles
- Application Roles

Permissions and Privileges



Want a great diagram that maps them all?

http://go.microsoft.com/fwlink/?LinkId=229142

Diagram taken from here where it is used as part of a general MSDN discussion on SQL Server permissions:

https://msdn.microsoft.com/en-us/library/ms191291.aspx

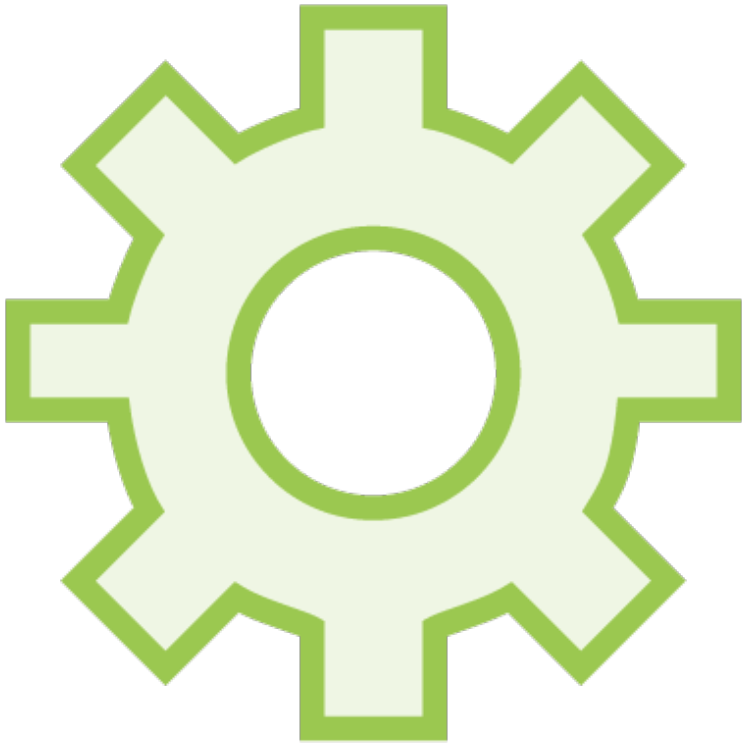
Principals

- Entities that can request resources

Privileges

- Permissions that can be granted or denied as they relate to resources

Securables



Securables

Any resource within SQL Server that has permissions or privileges that can be granted to a principal

SQL Server Authentication

Windows OS Login

Domain/Group Login

SQL Server Login

Windows OS and Domain Logins

Using domain based logins, especially with groups can ease administration by delegating maintenance to Active Directory

Windows OS and Domain Logins

When leveraging windows authentication SQL Server delegates “authentication” to the operating system or domain

Authentication

Process of establishing identity and ensuring validity of the login attempt

Windows authentication means this process has been delegated and SQL Server trusts the domain or OS

SQL Server Authentication

When using SQL Server authentication, SQL Server must use its own mechanisms to establish credentials of the connection

When Would You Use SQL Authentication

- Centralization of massive amounts of users
- Connections exist from outside domain
- Security architecture is application centric
- Rights are managed within the application

Considerations

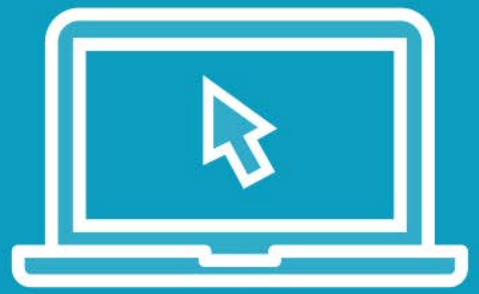
- Utilizing application centric security designs increases scope and complexity of audits

- Anonymous SQL Logins often have more rights than are needed by specific users

- Code should be secure, auditable, and controlled

- Application centric security designs are often susceptible to SQL Injection

Demo



Real World Demonstration

Next Up

Supporting Security With Relational Schema Design