ADMINISTRACIÓN DE LA SEGURIDAD DE BASE DE DATOS

Administración de usuarios, roles e inicios de sesión.

ING. JAIME LLANOS BARDALES

Introducción

- **Quiz**

Poll Everywhere

Objectives

Understand the differences between Windows, SQL Server and Azure Active Directory Authentication

Describe and configure both data-at-rest encryption solutions as well as data-in-transit encryption solutions

Implement a data sensitivity solution

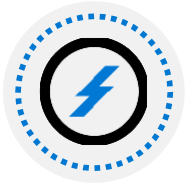
Objectives

Authentication options for Azure SQL Database

Security principals

Roles in Azure SQL Database

Understand permissions within Azure SQL



Understand the concept of least privilege

Azure AD authentication options

SQL Server authentication options	**Windows Authentication** User login information is stored in Active Directory
	SQL Server Authentication User login information is stored in the Master or user database
Azure SQL Database and Managed Instance authentication options	**Azure Active Directory Authentication** User information is stored in Azure Active Directory
	SQL Server Authentication User login information is stored in the master or user database

What's the difference between Active Directory and Azure Active Directory?

Universidad Nacional de Cajamarca
"Norte de la Universidad Peruana"

	Active Directory Domain Services	Azure Active Directory
User Management	Yes	Yes
Authentication	NTLM and Kerberos	OpenID Connect, SAML, OAuth
Groups	Yes	Yes
Object Hierarchy	Yes: X.500	Nope
Service Principals	Yes	Yes
Query AD programmatically	LDAP	AD Graph API (REST API)

Authentication and identities

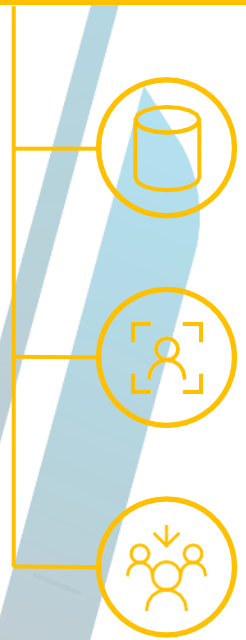
Authentication is the process of proving a user or service is who they say there are

Authorization is a process that occurs after a user is authenticated, and grants them their specified access to resources

Identities can represent users, service principals, or computers

Security principals

Security principals are any login, user, group, or role within the server or database



Users within the databases are either mapped to a login, or contained users within the database

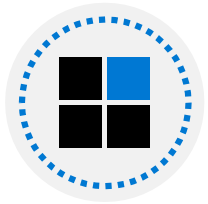
Contained users can be based on SQL Authentication or Azure Active Directory

Users can then be mapped to roles in order to give users centrally managed rights, or rights can be granted directly to a user

SQL Server Security overview

Securables

Object to which access must be secured

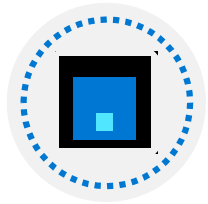


Principals

Security identities (users, service principals, or computers) that access securables to perform actions

Permissions

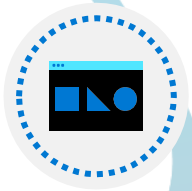
Actions principals can perform on securables



Security Hierarchies

Securables can contain other securables, and principals can contain other principals (roles)

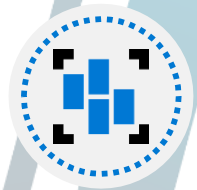
Schemas and securables



Securables are resources within databases like tables, views, procedures that access is granted to

Securable scopes: **<server>.<database>.<schema>.<object>**

Securables in Azure SQL Database only have the database and schema scopes

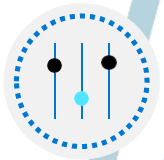


A schema is a collection of objects which allows objects to be grouped into separate namespaces

Logins and users

Logins are created in the master database and are used for server access

Instance level permissions are applied to logins

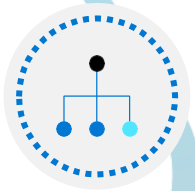


Database level permissions are applied to users

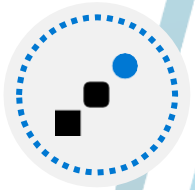
Contained users are authenticated at the database

A user has access only to the database in which they are created

Built-in database roles



SQL Server and Azure SQL Database include several fixed roles within each database

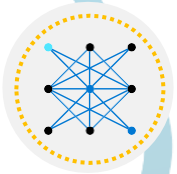


Users may be added as members of one or more roles (including custom roles)

The Master database in Azure SQL Database has a couple of unique roles since the *sysadmin* role does not exist

Server roles cannot be granted access to objects within a database directly and are only available in SQL Server and Azure SQL Managed Instance, but not in Azure SQL Database

Fixed server roles



Sysadmin

Can perform any activity on the server

Serveradmin

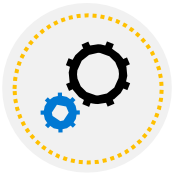
Can change server-wide configuration settings and can shutdown the server

Securityadmin

Can manage logins and their properties.

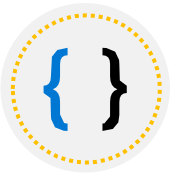
Processadmin

Can kill processes running inside of SQL Server



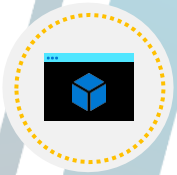
Setupadmin

Can add and remove linked servers using T-SQL



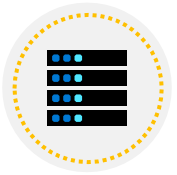
Bulkadmin

Can run the BULK INSERT T-SQL statement



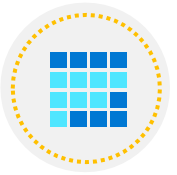
Diskadmin

Can manage backup devices in SQL Server



Dbcreator

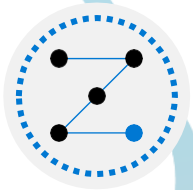
Can create, restore, alter, and drop any database



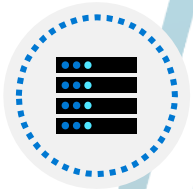
Public

Every SQL Server login belongs to the public user role.

Database roles in SQL Server vs. Azure SQL Database

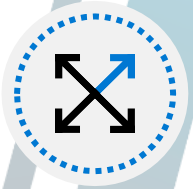


Roles are used to simplify the process of managing privileges in the database



In SQL Server and Managed Instance the scope of the role may be the **database** or the **server**

In Azure SQL Database roles are scoped to the individual **database**

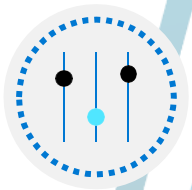


Both SQL Server and Azure SQL Database include built-in **database roles**, and allow for the creation of custom roles

Special Roles for Azure SQL Database

Database level roles available in the virtual master database only

---



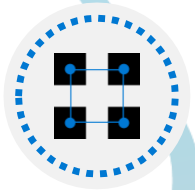
dbmanager – this role can create and delete databases. Equivalent to **dbcreator** fixed server role.

loginmanager – this role can create and delete logins in the virtual master database. Equivalent to **securityadmin** fixed server role.

Built-in Database Roles

- db_owner
- db_backupoperator
- db_datareader
- db_securityadmin
- db_ddladmin
- db_denydatawriter
- db_accessadmin
- db_datawriter
- db_denydatareader

Database and object permissions explained



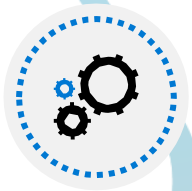
There are four **DML** permissions on tables and views – SELECT, INSERT UPDATE and DELETE

Stored Procedures and Functions have their own permissions – ALTER, CONTROL, EXECUTE, and VIEW DEFINITION

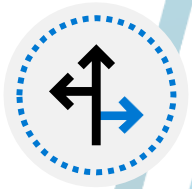
DCL permissions – GRANT, DENY, REVOKE
DDL permissions – CREATE, ALTER, DROP

Permissions which are REVOKED, remove any existing GRANT or DENY permission from the object

Database and object permissions explained



Permissions can be assigned to users or roles within a database



Users may then be assigned to roles

Permissions are additive, with permissions from multiple role memberships applied together

Preventing access through a DENY will override any GRANT to that object

GRANT / DENY example

```
GRANT SELECT ON dbo.Company to Demo
GO
DENY SELECT ON dbo.Company to Demo
GO
EXECUTE AS USER =  'Demo'

SELECT Name, Address FROM dbo.Company
```

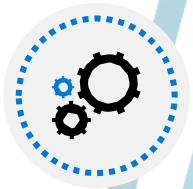
```
Messages
Msg 229, Level 14, State 5, Line 17
The SELECT permission was denied on the object 'Company', database 'WideWorldImporters', schema 'dbo'.

Completion time: 2020-05-13T14:42:28.8361616-07:00
```

EXECUTE AS USER / EXECUTE AS LOGIN definition

EXECUTE AS USER and EXECUTE AS LOGIN allows a statement to be executed in the security context of another user or login

---



This capability allows for **testing** during the development process to ensure permissions are correctly implemented

Fin de la sesión