

AZURE SQL

<https://vikasrajput.github.io>

Azure SQL is a family of fully managed, secure, and intelligent data service in cloud. Azure SQL DB was first launched in 2010, includes a serverless mode, and part of Azure SQL family.

HOW TO USE THIS FIELD MAP?

If new to Azure SQL, you can start with [Foundations](#), then check [References](#), and finally step through [WAF Pillars](#). Alternatively, feel free to jump through What's New or specific WAF [pillars](#).

FOUNDATIONS

[Pages and Extents](#)
[Index Architecture](#)
[Memory Management](#)
[Query Processing](#)
[Thread Architecture](#)
[Transaction Log](#)
[Transaction Locking](#)
[Execution Plans](#)
[Statistics](#)
[Query Profile](#)

Azure SQL service is available in two over-arching choices. Azure SQL DB which has further choices of Single DB, Elastic Pool, HyperScale and Serverless mode. And SQL Managed Instance (MI) which is enterprise-grade dedicated SQL DB as Service with committed capacity dedicated to respective customer only.

SQL DB: [Single Database](#), [Elastic Pool](#), [SQL Serverless](#), [Hyperscale](#)
SQL MI: [SQL Managed Instance](#)

Pricing: [vCore](#), [DTU](#), [Reserve](#), [Hybrid](#)
Tier: [General Purpose](#), [Business Critical](#)
[SQL DB vs MI Comparison](#)

[Multi model Features](#)
[In-Memory OLTP](#)
[Read Scaleout](#), [Scale Up/Down](#)
[Temporal Table - Ref + Retention](#)
[Planned Maintenance](#)

WHAT'S NEW

[Synapse link for SQL](#)
[Ledger](#)
[Elastic \[jobs, queries, transactions\]](#)
[Migrate from Hyperscale](#)
[Zone redundant Hyperscale](#)
[SQL Insights](#)
[SQL Analytics](#)
[SQL DB Emulator](#)

REFERENCES

[Azure SQL Status – Right Now!](#)
[Azure SQL – Committed SLA](#)
[Azure SQL by Region](#)

[Microsoft Learn Path – Azure SQL](#)
[YouTube Mechanics Channel - SQL](#)
[Azure Virtual Training Days - SQL](#)
[Microsoft SQL Server Blog](#)
[Microsoft Tech Community Channel](#)
[Managing Azure SQL – Free eBook](#)

[Microsoft Well Architected Framework](#)
[Cloud Data Migration Strategy Guide](#)
[Economics of Azure SQL Migration](#)
[App Modernization with Azure SQL](#)
[SQL Server Migration Enablement](#)
[Microsoft Partner Resources for SQL](#)

SECURITY

Security is the most critical WAF pillar. Let's get to it and categorize Security measures as Identity Management, Access Management, Surface Management, Protect Data and finally, Monitor/Audit.

[Azure Security Baseline + Policy Controls](#), [AAD](#), [MFA](#), [Managed Identity](#), [Policy](#), [Securables](#)
[Credential](#), [Principal](#), [Permission + Hierarchy](#), [Conditional Access + APIs](#), [RBAC: DB, App Encryption, Hierarchy](#), [Always Encrypted](#), [TDE](#), [RLS](#), [Data Masking](#), [Data Discovery](#)
[Defender for SQL](#), [Advanced Threat Protection](#), [Vulnerability Assessment](#)

SQL DB

[Ledger](#), [Security Enclaves](#), [DNS Alias](#)
[Connectivity Arch](#), [Network Controls](#)
[VNET Endpoint](#), [Private Link](#), [Firewall Rule](#), [Server Role](#), [SQL Audit](#)

SQL MI

[Instance Audit](#), [Public Endpoints](#)
[Server Trust Groups](#), [Linked Server](#)
[Windows Auth for AAD Principals](#)

COST OPTIMIZATION

After Security, generally Cost holds the next priority for clients. Though it's very difficult to talk about this pillar in isolation. Firstly, every Org needs to establish a baseline Consumption (capacity, cost) [Forecast](#), [Budget](#) and [Ownership](#) and then be able to track and [Report](#) consumption. And then, we need to approach Architecture as such to elevate Demand Management (e.g., throttle) and Supply Management (e.g., scale).

[Software Assurance to save on SQL Licenses](#), [Right Size](#), [Dev/Test](#), [Azure Backup Shift Capex to Opex](#), [Azure Hybrid](#), [Azure Advisor](#), [Cost Management](#), [Understand billing](#)

SQL DB

[Reserved Capacity](#), [Azure SQL Serverless](#), [SQL Elastic Pool](#)

SQL MI

[Reserved Instances](#), [Fine tune backups](#)

OPERATIONS EXCELLENCE

Ops Excellence proves the real-world agility and maturity of a business in managing Business Systems. At its core, Ops Excellence is all about how SDLC is managed, underscoring practices around Development, Deployment and Operation with Security, Monitoring and Automation embedded every step of the way.

[Database Lifecycle Management](#), [Data Compression: Row, Page, Unicode](#),
[SQL Server PowerShell](#), [Maintenance Window](#), [Azure SQL Analytics](#), [SQL Insights](#)
[CD with Azure DevOps \(lab\)](#), [DACPAC \(lab\)](#), [ARM Templates](#), [SSDT](#), [Retry Logic](#)

SQL DB

[Local Dev Experience](#), [Emulator](#), [Data Sync Agent](#),
[Move Shared Data](#)
[Database Sharding](#), [Shard Map](#), [Elastic Query](#),
[Query Routing](#)

SQL MI

[Resource Governor](#), [SQL Assessment API](#), [Managing Operations](#)
[Service Broker](#), [Database Mail](#)

RELIABILITY

Business should take lead on this and define must-have or preferred Availability and Recovery Metrics (SLA, MTTR, MTBF etc). This should inform Architecture – outlining High Availability (scale, prevent failure), Disaster Recovery (recognize failure, and recover) and Monitoring (service uptime, [chaos engineering](#), testing).

[Target SLA](#) and [Achieved SLA](#), [Accelerated DB Recovery](#), [Long-term Backup retention](#)
Recover with [Temporal tables](#), Plan and Develop for [Transient Errors](#)

SQL DB

[General \(local, zone-redundant\)](#), [Biz Critical \(local, zone-redundant\)](#)
[Hyperscale \(local, zone-redundant\)](#)
[Scale DB](#), [Pool resource](#), [Serverless](#)

SQL MI

[General \(local\)](#), [Biz Critical \(local\)](#)

PERFORMANCE

Performance is an interesting pillar – for it can be negatively impacted by almost all other pillars, but positive impact must be weaved in! Low-cost commitment or subpar operations or security measures can limit it. For this, establish Monitoring (workload, resources, baseline), Design for performance, and Remediate contention.

Monitoring. Most critical input in performance tuning is Baseline – workload & consumption. Workload expectation is business driven, and Consumption Baseline should be established (and adjusted) over a period.

[Batching for Performance](#), Optimize for [Memory](#), Config [App/DB](#) and [Server Log Diagnostic Telemetry](#), [Query Performance](#), [Query Store](#), [Intelligent Insights + flow](#)

SQL DB

[Query Perf Bottlenecks](#), [Query Tuning Assistant](#),
[Automatic Tuning](#)

SQL MI

[Query Performance Bottlenecks](#), [ML Service on SQL Managed Instance](#)