



# *Azure Data Services*

*- Azure SQL DB & Cosmos DB*

*Rajesh Kolla*

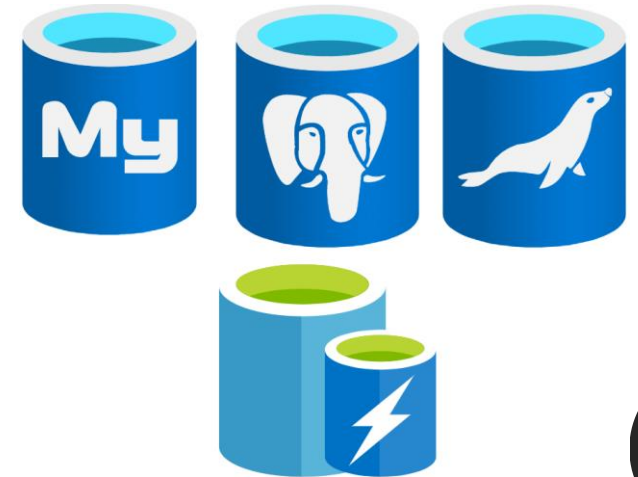
*Full-stack development , Azure Architect*

*Twitter: [@RajeshKolla18](https://twitter.com/RajeshKolla18)*

*LinkedIn: <https://be.linkedin.com/in/razeshkolla>*

# *Agenda*

- Types of Data
- Database Services in Azure
- Azure SQL DB Deployment options
- Azure SQL Database
- Demos
- Q&A



# Types of Data

## Structured data

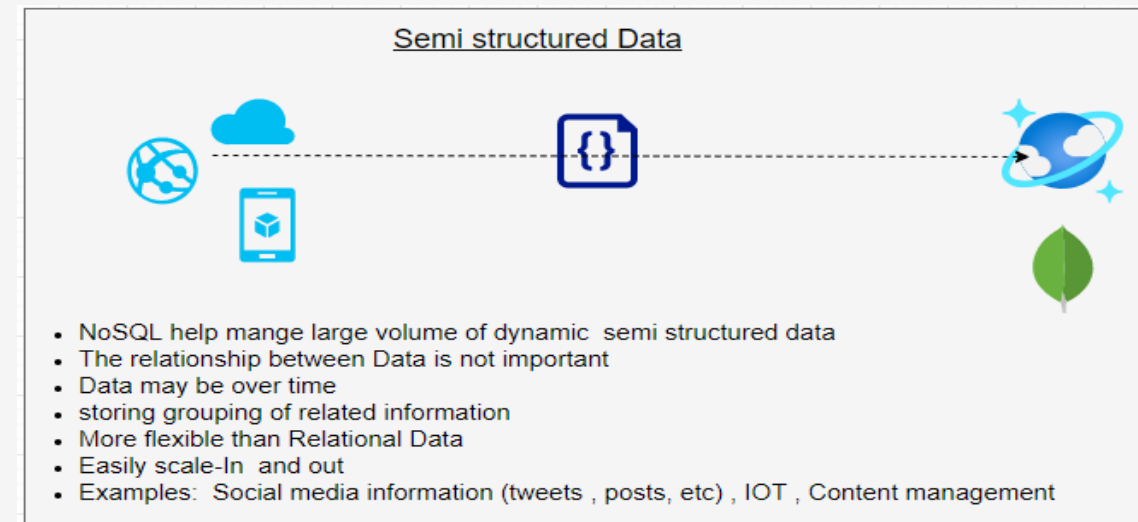
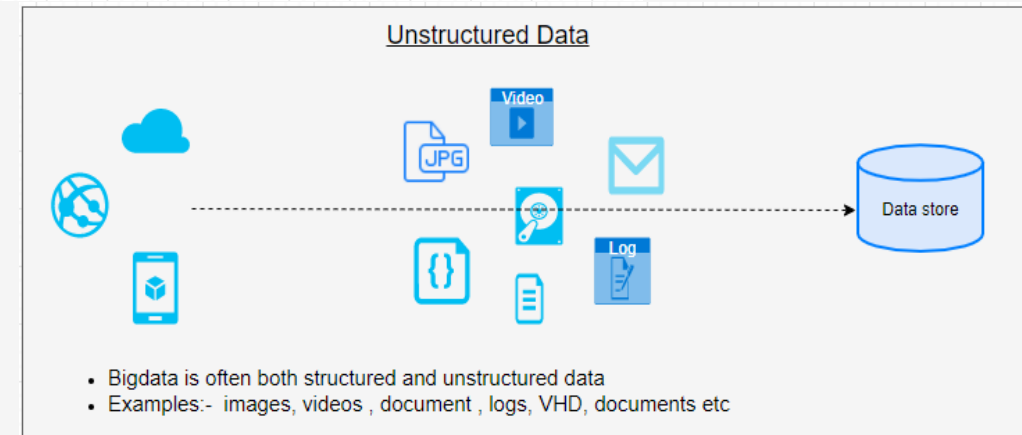
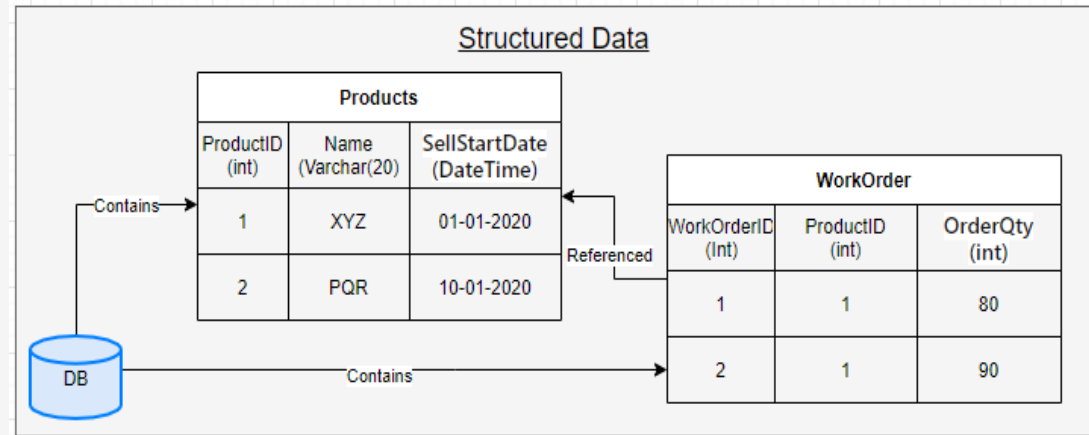
- Stick to schema
- Restriction on data to hold
- Stored in table with rows & columns
- Each row identified by Key
- Related Data between tables identified by Reference Key and also called "Relational data"

## Unstructured data

- Data is not Organized
- No restrictions on data to hold

## Semi structured data

- not completely structured but somewhat organized
- Uses keys \tags to organize data
- represent in JSON document \ Key value pair
- Referred as NoSQL or Non-relational Data



# Database services in Azure

| Database Service                                 | Use Case   |
|--|--|
| <a href="#">SQL Server on Virtual Machines</a>   | Lift and Shift existing On-Premises SQL workloads to Azure to maintain complete SQL Server compatibility and operating system-level access   |
| <a href="#">Azure SQL Managed Instance</a>       | Migrate existing SQL workloads to Azure complete SQL Server compatibility, Intelligent , scalable with all the benefits of a fully managed platform as a service   |
| <a href="#">Azure SQL Database</a>               | Build modern cloud applications with up-to-date relational database service with serverless compute, hyperscale storage, high availability with SLA 99.995% and AI-powered and automated features to optimize performance, threat detection and durability |
| <a href="#">Azure Cosmos DB</a>                  | Build applications with guaranteed low latency and high availability with SLA 99.999% anywhere, at any scale, or migrate Cassandra, MongoDB, and other NoSQL workloads to the cloud  |
| <a href="#">Azure Cache for Redis</a>            | Build fast and scalable applications with an open-source-compatible in-memory data store to cache database query results ,static content and session state distributed way by using <a href="#">cache aside pattern</a>                                    |
| <a href="#">Azure Database for MySQL</a>         | Deliver high availability with SLA up to 99.99% and elastic scaling to open-source mobile and web apps with a managed community MySQL database service, or migrate existing MySQL workloads to the cloud   |
| <a href="#">Azure Database for PostgreSQL</a>    | Build scalable, secure, and fully managed enterprise-ready apps on open-source PostgreSQL, scale out single-node PostgreSQL with high performance and availability up to 99.99% , or migrate existing PostgreSQL and Oracle workloads to the cloud         |
| <a href="#">Azure Database for MariaDB</a>       | Deliver high availability with SLA up to 99.99% and elastic scaling to open-source mobile and web apps with a managed community MariaDB database service or migrate existing MariaDB Workloads   |
| <a href="#">Azure Database Migration Service</a> | This is DB migration tool which simplify , automate database migration to azure.   |

Reference: <https://azure.microsoft.com/en-us/product-categories/databases/>



# Azure SQL

Relational Database-as-a- service

Azure SQL includes a range of products for hosting relational databases within Azure.

## Deployment Options for Azure SQL

1. Azure SQL virtual machines
2. Azure SQL managed instances
3. Azure SQL Databases



## What are the Primary Use case

- Build Applications with back end data repository
- Commonly used in Microsoft environments
- If need a fully managed SQL Database back-end

## SQL Virtual Machine

- Full Administrative control over the sqlserver
- Best for initial migrations (Lift & shift )to cloud with OS level access
- Expansive SQL Server and OS version Support
- Automated Manageability features for SQL Server

## SQL Managed Instances

- Intelligent, scalable cloud database service with all the benefits of a fully managed and platform as a service.
- Native Virtual network support
- Fully Managed service
- Enables frictionless migration for SQL workloads

## Features

- Provide Latest stable SQL Server engine
- Managed automated backups
- Database monitoring and metrics
- Automatic software patching
- Multiple data files per database
- single log file per database
- SSIS is part of Azure Data factory PaaS

## Pricing Model

support V-Core purchasing model (allow to change CPU, memory and Storage based on workload needs)

- **Gen 4** - Logical core based on Intel® E5-2673 v3 (Haswell), SSD , Physical cores , 7 GB Ram per core , Compute between 8-24 cores
- **Gen 5** - Logical cores based on Intel® E5-2673 v4 (Broadwell) 2.3 GHz. SSD , hyper threaded logical cores , 7 GB Ram per core , Compute between 8-24 cores



# Azure SQL

Relational Database-as-a- service

Azure SQL includes a range of products for hosting relational databases within Azure.

## Deployment Options for Azure SQL

1. Azure SQL virtual machines
2. Azure SQL managed instances
3. Azure SQL Databases



## SQL Databases

- Best suitable for modern cloud application
- Hyper scale storage up to 100 TB
- Server less Compute

### Types of Resources

#### Single Database

- Hyper scale storage up to 100 TB
- Serverless compute
- Easy Management

#### Elastic Pool

- Cost effective solution for cost optimization
- Resource sharing between multiple database
- Simplified performance solution

#### Database Server

\_used to manage group of single database and elastic pools

- Access management
- Backup Management
- Business Continuity Management

## Features for Azure SQL Database

- Uses Latest stable version of Microsoft SQL
- SQL elastic pools –economic resource sharing
- Serverless compute and Hyperscale storage
- Layers of protection , Intelligent threat protection
- Built-in AI for automatic performance tuning
- High Availability with SLA up to 99.995 %
- Failover Groups
- Geo-Replications
- Firewalls and virtual network support
- Dynamic data masking
- Transparent Data encryption at server
- Synch Groups
- Azure Search
- Create New Database or Migrate Existing Database using Microsoft Data Migration Service tool.



# Azure SQL :Pricing



## DTU Pricing

| Pricing  | CPU      | IOPS        | Latency               | Storage                      | Backups | Use case                                      |
|----------|----------|-------------|-----------------------|------------------------------|---------|---|
| Basic    | Low      | 5 DTU       | 5ms read<br>10ms read | DB <2 GB<br>per pool <156 DB | 7 days  | Small database with single concurrent user    |
| Standard | Low-High | 10-100 DTU  |                       | DB <1 TB<br>per pool <4 TB   | 35 days | Best for cloud apps with multiple operations  |
| Premium  | Med-High | 100-800 DTU | 2ms read<br>2ms read  | DB <4 TB<br>per pool < 4 TB  |         | High transaction volumes<br>Large no of users |

## vCore-based pricing

| Pricing           | Storage      | Latency                | Backups  | Backups   |
|-------------------|--------------|------------------------|--|---|
| General purpose   | 5 GB – 4 TB  | 5 -10 ms               | 7–35day backup<br>1 replica                        | Default option for most                                   |
| Business Critical | 5 GB – 4 TB  |                        | 7–35day backup<br>3 replicas , 1 read-scale backup | Business with I/O requirements                            |
| Hyperscale        | Up to 100 TB | Flexible –fast storage | Near-instant backups and fast restores             | Most workload and high scalable storage and fast restores |



# *Demo*

*Azure SQL Workloads*

---



# *Demo*

*Access Azure SQL Workloads using Entity Framework Core Packages*

---



# Cosmos DB

- Globally Distributed database
- Supports schema-less data
- Highly responsive with constantly changing data
- Multi model & multi master
- Designed for massive scale out
- Suitable for No SQL data
- Integration with Azure Function
- Azure Cognitive Search
- Serverless model without resource planning in advance
- Enterprise-grade security
- Encrypted at rest & transit
- Automatic Indexing

## Cosmos DB API Options & Consistency levels

- SQL (Core)
- Cassandra
- MongoDB
- Gremlin
- Table

| Consistency       | Details   |
|-------------------|---|
| Strong            | Guaranteed write operation only committed and visible on the primary after it has been committed and confirmed by all replicas.   |
| Bounded Staleness | Allows to configure how stale docs can be within replicas; staleness means the quantity or version count a replica document can be behind a primary document.                                   |
| Session           | Guarantees that all read and write operations are consistent within a user session.   |
| Consistent Prefix | Guarantees changes are read in the order that matches the sequence of the corresponding writes.   |
| Eventual          | Offers looser consistency and commits and write operations against the primary immediately. Replica transactions are asynchronously handled and will eventually be consistent with the primary. |



# Cosmos DB :Pricing



- Operation of Cosmos DB measures in Request Unit (RU) and billed per hour
- Storage billed in GB by SSD . Two backup copies are free & additional copies are charged in GB of data stored.
- There are mainly two types of pricing model
  - ❖ Provisioned Throughput
    - ❖ Standard
      - Guaranteed low latency and high availability with SLA 99.9999 %
      - Single digit milliseconds reads and writes
      - ideal for large & Critical workloads
      - starting minimum 400 RU/sec
    - ❖ Autoscale
      - No need to manually manage capacity of large & critical workloads (Additional feature over standard throughput )
      - Custom throughput limit starting 4000 RU/sec
  - ❖ Serverless
    - Eliminates over-provision the database infrastructure
- Save up to 65% of billing cost with [reserved capacity](#)
- Save up to 70% of billing cost with [TCO](#)



# *Demo*

*No SQL workloads using Cosmos DB*

---

# *Demo*

*Accessing Cosmos DB data using Cosmos DB using Entityframework core*

---

# *Third-Party Managed Databases in Azure*

- Azure Database for MySQL
- Azure Database for PostgreSQL
- Azure Database for MariaDB



## Managed Database Options

- Build-in high availability with SLA up to 99.99%
- Auto-scaling
- Encrypt at rest and in –transit
- Automatic backups with 35 days retention period
- Pay as you go
- High predictable performance



# *Demo*

*Third party Managed Database workloads*

---

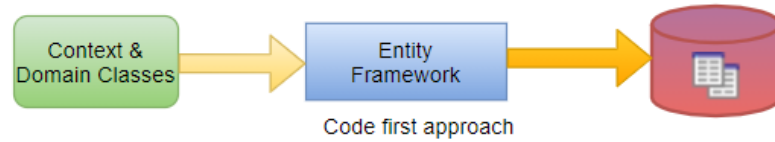
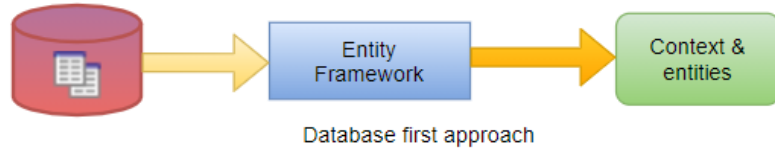
# *Demo*

*Third party Managed Database workloads using Entityframework core*

---



# Entity Framework Core



- Entity Framework Core is the version of Entity Framework after EF 6.x.
- Open-source, lightweight, extensible and a cross-platform version of Entity Framework data access technology.
- An Object/Relational Mapping (O/RM) framework.
- EF Core is used with .NET Core applications. , it can also be used with standard .NET 4.5+ framework based applications.
- [Database first approach](#) : Generate Data access classed from existing Database
- [Database first approach](#) : Generate Data access classed from existing Database
- [Microsoft.EntityFrameworkCore](#)
- [Microsoft.EntityFrameworkCore.Design](#)

Entity framework Core uses Provider model to access many different databases

- SQLServer - [Microsoft.EntityFrameworkCore.SqlServer](#)
- MySQL - [MySQL.Data.EntityFrameworkCore](#)
- PostgreSQL - [Npgsql.EntityFrameworkCore.PostgreSQL](#)
- SQLite - [Microsoft.EntityFrameworkCore.SQLite](#)
- SQLCompact - [EntityFrameworkCore.SqlServerCompact40](#)
- In-memory - [Microsoft.EntityFrameworkCore.InMemory](#)
- Cosmos - [Microsoft.EntityFrameworkCore.Cosmos](#)

# *Q&A*

---



*Thank you!*

Twitter: @RajeshKolla18

LinkedIn: <https://be.linkedin.com/in/razeshkolla>