Design and implement cloud-native applications with Microsoft Azure Cosmos DB Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Overview

What Can You Achieve with This Training?

- Learn to build **high-performance, scalable cloud-native applications** using **Azure Cosmos DB for NoSQL**.
- Develop reliable data access solutions with **SDKs, data operations, querying, and data modelling** techniques.
- Balance performance and cost through effective **partitioning, throughput, consistency, and indexing** strategies.
- Explore **full-text search, vector search, and hybrid search** for modern AI-powered applications.
- Build **Retrieval-Augmented Generation (RAG)** solutions and persistent **AI agent memory stores** using Azure Cosmos DB.
- Understand how **Microsoft Entra, RBAC, backup, restore, and multi-region configurations** support secure and available database solutions.
- Learn to monitor and manage data environments with **Azure Monitor, Cosmos DB Fleets, and Microsoft Fabric**.

The **Design and Implement Cloud-Native Applications with Microsoft Azure Cosmos DB Training** helps developers design, build, and optimise **scalable, high-performance cloud-native and AI-enabled applications** using **Azure Cosmos DB for NoSQL**.

The course introduces Azure Cosmos DB as both a NoSQL database service and a data platform that can support modern AI application scenarios.

Participants learn to evaluate whether **Azure Cosmos DB for NoSQL** meets application requirements, create accounts, databases, containers, and items, and configure resources for different workloads. Topics such as **Request Units (RUs), Throughput, Consistency, and Time to Live (TTL)** are examined in relation to performance, data retention, and cost.

The programme covers application connectivity using **.NET and Python SDKs**, including point reads, item creation and updates, deletion, partial document patching, transactional batches, bulk operations, and optimistic concurrency control.

Data architecture modules focus on **access-pattern-based data modelling, partition key selection, hierarchical and synthetic partitioning, multitenant strategies, and indexing policies**.

Participants also explore **Change Feed processing**, integration with **Azure Functions**, and moving data through **Container Copy Jobs**.

The AI-focused modules introduce **Model Context Protocol (MCP)**, AI-assisted development tools, **full-text search, vector search, BM25 ranking, Reciprocal Rank Fusion (RRF), and hybrid retrieval**.

Design and implement cloud-native applications with Microsoft Azure Cosmos DB Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

Learners then examine how Azure Cosmos DB can support **Retrieval-Augmented Generation (RAG)** applications and persistent memory solutions for AI agents.

Security and availability topics include **Microsoft Entra authentication, role-based access control, network isolation, dynamic data masking, backup and restore, point-in-time recovery, and multi-region failover**.

The course also covers **query optimisation, monitoring with Azure Monitor, managing Cosmos DB environments at scale with Fleets**, and integrating operational data with **Microsoft Fabric OneLake**.

Throughout the programme, **module assessments and practical exercises** help participants apply the concepts to realistic application development and database management scenarios.

Prerequisites

Participants should have:

- Familiarity with **Microsoft Azure fundamentals**
- Development experience with **C#, Java, Python, or JavaScript**
- Experience working with **SQL or NoSQL databases**
- A basic understanding of **application development concepts**

Who Should Attend

This course is intended for developers who build or support **cloud-native applications using Azure Cosmos DB for NoSQL**.

It is particularly suitable for:

- **Software Developers**
- **Application Developers**
- **Cloud Developers**
- **Solutions Developers**
- Professionals developing **modern cloud-native applications**

Design and implement cloud-native applications with Microsoft Azure Cosmos DB Training

Duration: **4 Day**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

What You Will Learn

By the end of this course, participants will be able to:

- Evaluate **Azure Cosmos DB for NoSQL** for different application and AI workloads.
- Create and manage **accounts, databases, containers, and items**.
- Configure **Request Units, Throughput, Consistency, and TTL** to balance performance and cost.
- Connect to Azure Cosmos DB using **.NET and Python SDKs**.
- Perform **point reads, create, replace, upsert, delete, patch, transactional batch, and bulk operations**.
- Apply **optimistic concurrency control** to data updates.
- Write **parameterised queries, correlated subqueries, and cross-product queries**.
- Page large query results using **continuation tokens**.
- Design data models and select **partitioning strategies** based on access patterns.
- Evaluate **hierarchical, synthetic, and multitenant partitioning** approaches.
- Process database changes using the **Change Feed** and **Azure Functions**.
- Use **MCP Server and AI-assisted development tools** in Azure Cosmos DB scenarios.
- Secure solutions using **Microsoft Entra, RBAC, network isolation, and dynamic data masking**.
- Configure **periodic backup, continuous backup, and point-in-time recovery**.
- Evaluate **multi-region availability, multi-region writes, conflict resolution, and failover** strategies.
- Develop indexing strategies using **composite, spatial, computed-property, and global secondary indexes**.
- Analyse **query cost, latency, scans, fan-out operations, and partition utilisation**.
- Configure **Azure Monitor alerts** and investigate diagnostic logs.
- Manage multitenant environments and assess usage and cost through **Cosmos DB Fleets**.
- Implement **full-text search, vector search, BM25 ranking, and vector similarity**.
- Develop **hybrid search** solutions using **Reciprocal Rank Fusion (RRF)**.
- Build **Retrieval-Augmented Generation (RAG)** applications with contextual retrieval and grounded responses.
- Design **short-term and long-term memory stores for AI agents**.

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Use **Microsoft Fabric OneLake**, **T-SQL**, and **Spark** for operational analytics on Azure Cosmos DB data.

Outline

1. Explore Azure Cosmos DB for NoSQL

This module introduces the core resources and capabilities of Azure Cosmos DB for NoSQL.

- Evaluate **Azure Cosmos DB suitability** for application and AI workloads
- Create **accounts, databases, containers, and items**
- Explore **Cosmos DB resources and basic operations**
- **Exercise:** Create an Azure Cosmos DB account and resources

2. Configure Resources, Throughput, and Consistency

This module examines how to configure Azure Cosmos DB capacity and consistency to meet application requirements.

- Assess **Request Unit (RU)**, **Throughput**, and **Consistency** requirements
- Configure **provisioned throughput** and **serverless capacity**
- Configure **throughput and consistency levels**
- Set **Time to Live (TTL)** for data retention
- Balance **cost and performance**
- **Exercise:** Configure throughput and consistency

3. Connect to Azure Cosmos DB with the SDK

This module focuses on application connectivity and SDK configuration.

- Initialise and configure **.NET and Python SDKs**
- Connect using a **client singleton**
- Tune **connectivity modes**
- Enable **offline development**
- Handle **connection errors**

4. Implement Azure Cosmos DB Operations with the SDK

This module introduces the data operations available through Azure Cosmos DB SDKs.

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Perform **point read operations**
- **Create, replace, upsert, and delete items**
- Update items using **partial document patch**
- Configure **item-level Time to Live (TTL)**
- Implement **optimistic concurrency control**
- Execute **transactional batch operations**
- Perform **bulk operations**
- **Exercise:** Implement data operations

5. Query Data in Azure Cosmos DB for NoSQL

This module explores advanced query techniques and result management.

- Write **parameterised queries**
- Use **projections and built-in functions**
- Implement **correlated subqueries**
- Implement **cross-product queries**
- Page large result sets using **continuation tokens**
- **Exercise:** Author and page queries

6. Design a Data Modeling and Partitioning Strategy

This module examines data modelling and partitioning approaches for scalable workloads.

- Design data models based on **access patterns**
- Select **partition keys** for scalable applications
- Choose **hierarchical partitioning**
- Evaluate **synthetic partitioning**
- Apply **multitenant partitioning** strategies
- Use **denormalisation and pre-aggregation** for performance
- Identify **data modelling anti-patterns**
- Estimate **workload cost**
- **Exercise:** Design a data model and partitioning strategy

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

7. Process the Azure Cosmos DB for NoSQL Change Feed

This module focuses on processing database changes and supporting event-driven applications.

- Read the **Change Feed** using processor and pull models
- Select an appropriate **Change Feed mode**
- Respond to changes using **Azure Functions**
- Move container data using **Container Copy Jobs**
- **Exercise:** Process changes and copy data

8. Implement AI-Assisted Development Tools for Azure Cosmos DB

This module introduces AI-assisted development tools and their integration with Azure Cosmos DB.

- Connect an AI assistant through a **Model Context Protocol (MCP) Server**
- Add **durable memory** using agent toolkits
- Implement **multi-step retrieval**
- Select appropriate tools for different scenarios
- **Exercise:** Inspect Azure Cosmos DB using **Azure MCP Server**

9. Secure an Azure Cosmos DB Solution

This module covers identity, access management, and data protection controls.

- Configure **Microsoft Entra authentication**
- Configure **key-based authentication**
- Implement **Role-Based Access Control (RBAC)**
- Set up **network isolation**
- Apply **dynamic data masking**
- **Exercise:** Secure an Azure Cosmos DB account

10. Configure Backup and Restore in Azure Cosmos DB

This module introduces backup options and recovery strategies.

- Compare **periodic and continuous backup**
- Enable **continuous backup**
- Perform **Point-in-Time Recovery (PITR)**

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- Describe **Azure Backup** options for long-term retention
- **Exercise:** Recover data from a recovery point

11. Design Multi-Region Availability and Failover in Azure Cosmos DB

This module examines high availability and regional deployment configurations.

- Add **regions** to an Azure Cosmos DB account
- Route **reads and writes** across regions
- Configure **multi-region writes**
- Implement **conflict resolution**
- Configure **account-level failover**
- Configure **partition-level failover**
- **Exercise:** Configure regions and failover

12. Design and Optimize an Indexing Strategy in Azure Cosmos DB

This module explores indexing approaches for different query and workload requirements.

- Align **indexing policies** with workload requirements
- Configure **composite indexes**
- Configure **spatial indexes**
- Configure **computed-property indexes**
- Implement **global secondary indexes** for cross-partition queries
- **Exercise:** Optimise an indexing strategy

13. Analyze and Tune Query and Operation Performance in Azure Cosmos DB

This module focuses on identifying expensive queries and improving database performance.

- Diagnose **high-cost queries** using metrics
- Distinguish **scans** from **fan-out operations**
- Apply appropriate **performance remediation techniques**
- Analyse **query performance metrics**
- Evaluate **index performance metrics**
- Optimise **cross-partition queries**

BİLGİNÇ IT Academy

info@bilginc.com

bilginc.com

- **Exercise:** Tune a high-cost query

14. Monitor and Troubleshoot Azure Cosmos DB

This module introduces monitoring and troubleshooting approaches for Azure Cosmos DB workloads.

- Interpret **response codes**
- Diagnose **throttling**
- Monitor **latency**
- Monitor **partition utilisation**
- Configure **Azure Monitor alerts**
- Analyse **diagnostic logs**
- **Exercise:** Diagnose throttling and configure an alert

15. Manage Azure Cosmos DB at Scale with Fleets

This module examines centralised management of large and multitenant Cosmos DB environments.

- Group accounts into a **Fleet**
- Manage a **multitenant estate** as a single unit
- Share throughput through **pools**
- Monitor **Fleet Analytics**
- Analyse **estate-wide usage and costs**
- **Exercise:** Create and monitor a Fleet

16. Implement Full-Text and Vector Search in Azure Cosmos DB for NoSQL

This module explores native keyword and semantic search capabilities.

- Add **keyword retrieval** to containers
- Implement **semantic retrieval**
- Configure **full-text search policies**
- Configure **vector policies and indexes**
- Select appropriate **partitioning** for search workloads
- Rank results using **BM25**
- Apply **vector similarity** for semantic ranking

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Use the **Change Feed** to keep embeddings aligned with source text
- **Exercise:** Build native search over a product catalogue

17. Implement Hybrid Search and Optimize AI Retrieval in Azure Cosmos DB for NoSQL

This module focuses on combining retrieval techniques and improving search performance.

- Combine **keyword and semantic retrieval**
- Apply **Reciprocal Rank Fusion (RRF)**
- Tune **result relevance**
- Evaluate **request charge**
- Optimise **latency**
- Isolate **search workloads** from transactional traffic
- **Exercise:** Build and tune hybrid search

18. Build RAG Applications with Azure Cosmos DB for NoSQL

This module introduces Retrieval-Augmented Generation applications supported by Azure Cosmos DB.

- Prepare content for **retrieval**
- Generate **embeddings**
- Retrieve relevant **context**
- **Ground model responses** in retrieved information
- Assemble **prompts that cite sources**
- Handle **evidence gaps**
- Orchestrate a **RAG pipeline** for follow-up questions
- **Exercise:** Build a grounded RAG assistant

19. Design and Implement Agent Memory Stores in Azure Cosmos DB

This module explores persistent memory architectures for AI agents.

- Separate **short-term conversation state** from **long-term memory**
- Distil raw conversation turns into facts for later retrieval
- Configure **retention policies**
- Apply **reconciliation policies**

BİLGİNÇ IT Academy
info@bilginc.com
bilginc.com

- Define **erasure policies**
- **Exercise:** Build a durable agent memory store

20. Implement Operational Analytics with Microsoft Fabric

This module examines how Azure Cosmos DB data can support analytics in Microsoft Fabric.

- Mirror an Azure Cosmos DB database into **Microsoft Fabric OneLake**
- Verify the **mirrored data copy**
- Query mirrored data using **T-SQL**
- Analyse mirrored data using **Apache Spark**
- Monitor **mirroring operations**
- Answer **operational and analytical questions**
- **Exercise:** Mirror and query a Cosmos DB database in Microsoft Fabric