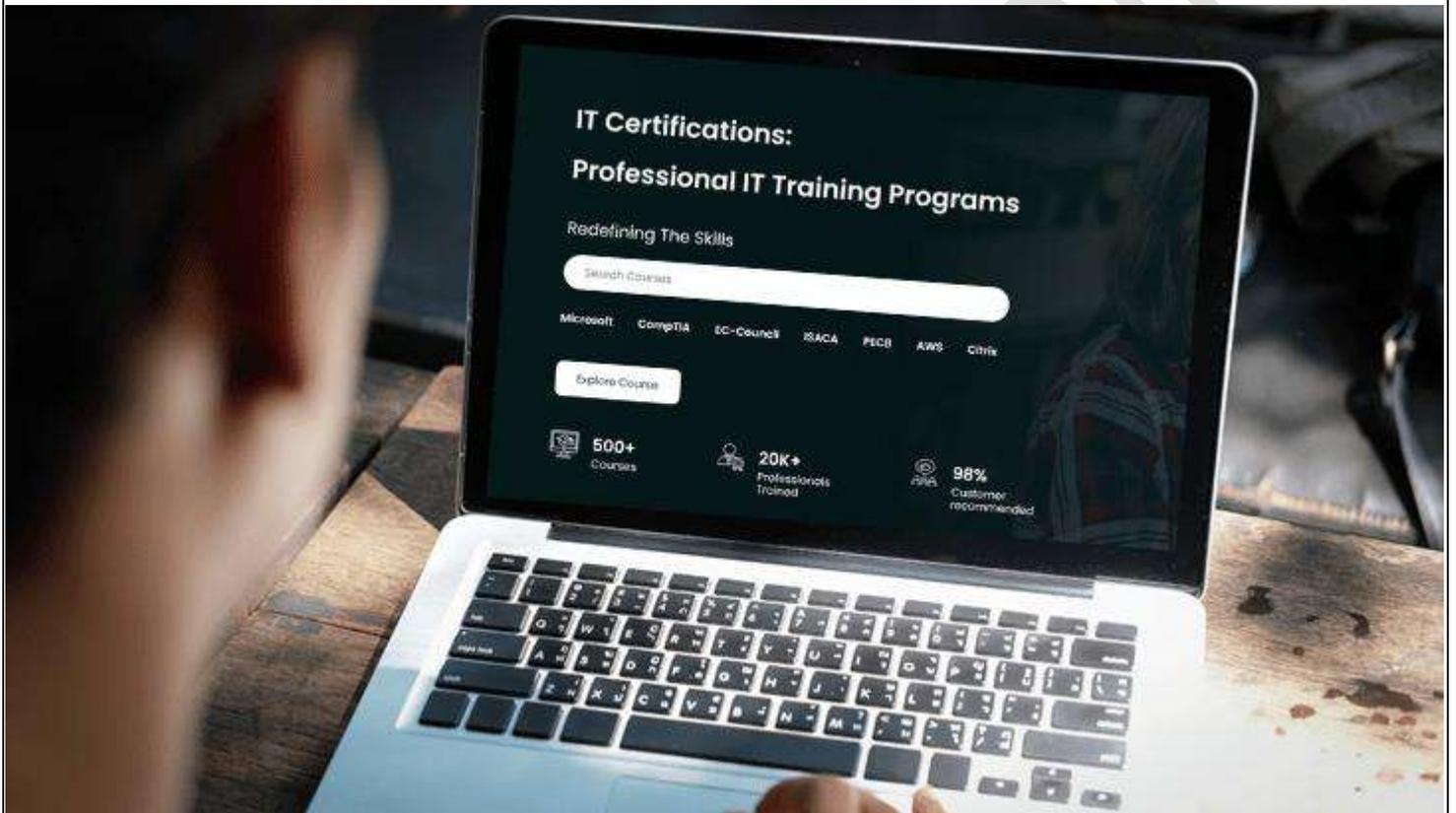




Redefining The Skills



## **DP-420T00: DESIGNING AND IMPLEMENTING CLOUD-NATIVE APPLICATIONS USING MICROSOFT AZURE COSMOS DB TRAINING**

**Duration: 4 Days**

### **Course Description**

DP-420T00: Designing and Implementing Cloud-Native Applications using Microsoft Azure Cosmos DB Training is the cloud-native application using Microsoft Cosmos DB Training. This course will help you learn about how to build an application with the help of SQL API & SDK for Cosmos DB Azure.

In this particular course structure candidates will also learn to write queries, manage & provision resources, creating index policies and executing everyday operations by using SDKs.

### **Who should attend this course?**

- Engineers whose job is authoring cloud-native solutions that influence Cosmos DB Azure SQL API and its many SDKs.
- Professionals who know Python, C#, JavaScript, and Java
- Professionals who have expertise in writing code with a SQL & NoSQL database platform.
- Given below are professionals who can use Designing and Implementing Cloud-Native Applications Using Microsoft Azure Cosmos DB Training to upskill their current positions:
  - Data Engineers
  - DevOps Engineers
  - System Administrators
  - Data Scientist
  - Application Developers
  - Database Administrators

### **What you will learn**

- Design and configure Cosmos DB Azure SQL API, container, and database.
- Utilize the .NET SDK to operate resources and execute operations.
- Execute queries of changing complexity.
- Create a data model and partitioning technique.
- Optimize questions and indexing based on attributes of an application.
- Utilizing the Azure Resource Manager to operate accounts and resources with JSON or CLI and Bicep templates.

### **Prerequisites**

- Understanding of Microsoft Azure and the capability to guide the Azure portal (equivalent to AZ-900) is a must
- Knowledge of writing in Azure-supported language at a middle level (Javascript, C#, Java, or Python) is a compulsion
- To be able to write code that combines and conducts operations on NoSQL or SQL database products. (Oracle, SQL Server, Cassandra, MongoDB, or similar) is a must
- AZ-900T00: Microsoft Azure Fundamentals

### **Curriculum**

#### **Module 1: Get started with Azure Cosmos DB for NoSQL**

- Introduction to Azure Cosmos DB for NoSQL
  - Evaluate whether Azure Cosmos DB for NoSQL is the right database for your application.

- Describe how the features of the Azure Cosmos DB for NoSQL are appropriate for modern applications.
- Try Azure Cosmos DB for NoSQL
  - Create a new Azure Cosmos DB for NoSQL account
  - Create database, container, and item resources for an Azure Cosmos DB for NoSQL account

#### **Module 2: Plan and implement Azure Cosmos DB for NoSQL**

- Plan Resource Requirements
  - Evaluate various requirements of your application
- Configure Azure Cosmos DB for NoSQL
  - Compare the various service and throughput offerings for Azure Cosmos DB
  - Migrate between standard and autoscale throughput
- Move data into and out of Azure Cosmos DB for NoSQL
  - Migrate data using Azure services
  - Migrate data using Spark or Kafka

#### **Module 3: Connect to Azure Cosmos DB for NoSQL with the SDK**

- Use the Azure Cosmos DB for NoSQL SDK
  - Integrate the Microsoft.Azure.Cosmos SDK library from NuGet
  - Connect to an Azure Cosmos DB for NoSQL account using the SDK and .NET
- Configure the Azure Cosmos DB for NoSQL SDK
  - Configure the SDK for offline development
  - Troubleshoot common connection errors
  - Implement parallelism in the SDK
  - Configure logging using the SDK

#### **Module 4: Access and manage data with the Azure Cosmos DB for NoSQL SDKs**

- Implement Azure Cosmos DB for NoSQL point operations
  - Perform CRUD operations using the SDK
  - Configure TTL for a specific item
- Perform cross-document transactional operations with the Azure Cosmos DB for NoSQL
  - Create a transactional batch and review results
  - Implement optimistic concurrency control for an operation
- Process bulk data in Azure Cosmos DB for NoSQL
  - Use C# task asynchronous programming model and "bulk" support in the Azure Cosmos DB for NoSQL .NET SDK

#### **Module 5: Execute queries in Azure Cosmos DB for NoSQL**

- Query the Azure Cosmos DB for NoSQL
  - Create and execute a SQL query
  - Project query results
  - Use built-in functions in a query
- Author complex queries with the Azure Cosmos DB for NoSQL
  - Implement a correlated subquery
  - Create a cross-product query

#### **Module 6: Define and implement an indexing strategy for Azure Cosmos DB for NoSQL**

- Define indexes in Azure Cosmos DB for NoSQL
  - Create and execute a SQL query
  - Project query results
  - Use built-in functions in a query

- Customize indexes in Azure Cosmos DB for NoSQL
  - Implement a correlated subquery
  - Create a cross-product query

#### **Module 7: Integrate Azure Cosmos DB for NoSQL with Azure services**

- Consume an Azure Cosmos DB for NoSQL change feed using the SDK
  - Create a change feed processor in the .NET SDK
  - Author a delegate to handle a batch of changes in a client-side application
- Handle events with Azure Functions and Azure Cosmos DB for NoSQL change feed
  - Create an Azure Function using the Azure Cosmos DB trigger
- Search Azure Cosmos DB for NoSQL data with Azure Cognitive Search
  - Create an indexer to migrate data from Azure Cosmos DB for NoSQL to an Azure Cognitive Search index

#### **Module 8: Implement a data modeling and partitioning strategy for Azure Cosmos DB for NoSQL**

- Implement a non-relational data model
  - Determine access patterns for data.
  - Apply data model and partitioning strategies to support an efficient and scalable NoSQL database.
- Design a data partitioning strategy
  - Manage relationships between data entities by using advanced modeling and partitioning strategies.
  - Maintain the referential integrity of your data by using change feed.
  - Implement pre-aggregating and denormalizing data strategies to improve data-model performance and scaling.
  - Optimizing storage and compute by mixing entity types in a single container

#### **Module 9: Design and implement a replication strategy for Azure Cosmos DB for NoSQL**

- Configure replication and manage failovers in Azure Cosmos DB
  - Distribute data across global data centers
  - Configure automatic failover and perform a manual failover
  - Configure the Azure Cosmos DB for NoSQL .NET SDK to use a specific region
- Use consistency models in Azure Cosmos DB for NoSQL
  - Configure default consistency level for an Azure Cosmos DB account
  - Change the consistency level on a per-request basis
- Configure multi-region write in Azure Cosmos DB for NoSQL
  - Configure Azure Cosmos DB for multi-region write
  - Use the .NET SDK to select a write region for operations.

#### **Module 10: Optimize query and operation performance in Azure Cosmos DB for NoSQL**

- Customize an indexing policy in Azure Cosmos DB for NoSQL
  - Customize an indexing policy for read-heavy workloads
  - Customize an indexing policy for write-heavy workloads
- Measure index performance in Azure Cosmos DB for NoSQL
  - Tune an indexing policy for specific queries
  - Measure the cost for a query or operation
- Implement integrated cache in Azure Cosmos DB for NoSQL
  - Implement the integrated cache
  - Configure integrated cache options

### **Module 11: Monitor and troubleshoot an Azure Cosmos DB for NoSQL solution**

- Measure performance in Azure Cosmos DB for NoSQL
  - Understand how Azure Cosmos DB uses Azure Monitor to monitor server-side metrics.
  - Measure Cosmos DB's throughput
  - Observe rate-limiting events.
  - Query telemetry logs
  - Measure cross-partition storage distribution throughput
- Monitor responses and events in Azure Cosmos DB for NoSQL
  - Review common response codes
  - Understand transit errors
  - Review rate-limiting errors
  - Configure Alerts
  - Audit Security
- Implement backup and restore for Azure Cosmos DB for NoSQL
  - Understand the different backup and restore options Azure Cosmos DB provides.
  - Evaluate periodic backups.
  - Configure continuous backups.
  - Do point in time recovery.
- Implement security in Azure Cosmos DB for NoSQL
  - Implement network level access control
  - Review data encryption options
  - Use role-based access control (RBAC)
  - Access Account Resources using Microsoft Entra ID
  - Understand Always Encrypted

### **Module 12: Manage an Azure Cosmos DB for NoSQL solution using DevOps practices**

- Write management scripts for Azure Cosmos DB for NoSQL
  - View arguments, groups, and subgroups for a specific CLI command
  - Create Azure Cosmos DB accounts, databases, and containers using the CLI
  - Manage an indexing policy using the CLI
  - Configure database or container throughput using the CLI
  - Initiate failovers and manage failover regions using the CLI
- Create resource template for Azure Cosmos DB for NoSQL
  - Identify the three most common resource types for Azure Cosmos DB for NoSQL accounts
  - Create and deploy a JSON Azure Resource Manager template for an Azure Cosmos DB for NoSQL account, database, or container
  - Create and deploy a Bicep template for an Azure Cosmos DB for NoSQL account, database, or container
  - Manage throughput and index policies using JSON or Bicep templates

### **Module 13: Create server-side programming constructs in Azure Cosmos DB for NoSQL**

- Build multi-item transactions with the Azure Cosmos DB for NoSQL
  - Author stored procedure
  - Rollback stored procedure transactions
- Expand query and transaction functionality in Azure Cosmos DB for NoSQL
  - Create user-defined functions
  - Create triggers

## LAB Outline

- Create lab resource group
- Setup lab environment
- Enable resource providers
- Create an Azure Cosmos DB for NoSQL account
- Configure throughput for Azure Cosmos DB For NoSQL with the Azure portal
- Migrate existing data using Azure Data Factory
- Connect to Azure Cosmos DB for NoSQL with the SDK
- Configure the Azure Cosmos DB for NoSQL SDK for offline development
- Create and update documents with the Azure Cosmos DB for NoSQL SDK
- Batch multiple point operations together with the Azure Cosmos DB for NoSQL SDK
- Move multiple documents in bulk with the Azure Cosmos DB for NoSQL SDK
- Execute a query with the Azure Cosmos DB for NoSQL SDK
- Paginate cross-product query results with the Azure Cosmos DB for NoSQL SDK
- Review the default index policy for an Azure Cosmos DB for NoSQL container with the portal
- Process change feed events using the Azure Cosmos DB for NoSQL SDK
- Process Azure Cosmos DB for NoSQL data using Azure Functions
- Search data using Azure AI Search and Azure Cosmos DB for NoSQL
- Measure performance for customer entities
- Cost of denormalizing data and aggregates and using the change feed for referential integrity
- Connect to different regions with the Azure Cosmos DB for NoSQL SDK
- Configure consistency models in the portal and the Azure Cosmos DB for NoSQL SDK
- Connect to a multi-region write account with the Azure Cosmos DB for NoSQL SDK
- Optimize an Azure Cosmos DB for NoSQL container's indexing policy for common operations
- Optimize an Azure Cosmos DB for NoSQL container's index policy for a specific query
- Use Azure Monitor to analyze an Azure Cosmos DB for NoSQL account
- Troubleshoot an application using the Azure Cosmos DB for NoSQL SDK
- Recover a database or container from a recovery point
- Store Azure Cosmos DB for NoSQL account keys in Azure Key Vault
- Adjust provisioned throughput using an Azure CLI script
- Create an Azure Cosmos DB for NoSQL container using Azure Resource Manager templates
- Create a stored procedure with the Azure portal
- Implement and then use user-defined functions with the SDK

---

*For any query Contact Us – Microtek Learning*

---