

Training: Microsoft
DP-420T00 Designing and Implementing Cloud-Native Applications Using
Microsoft Azure Cosmos DBMicrosoft
Partner

TRAINING GOALS:

This course teaches developers how to create application using the SQL API and SDK for Azure Cosmos DB. Students will learn how to write efficient queries, create indexing policies, manage and provisioned resources, and perform common operations with the SDK.

Skills gained:

- Create and configure Azure Cosmos DB SQL API account, database, and container
- Use the .NET SDK to manage resources and perform operations
- Perform queries of varying complexity
- Design a data modeling and partitioning strategy
- Optimize queries and indexes based on characteristics of an application
- Use the Azure Resource Manager to manage accounts and resources with CLI or JSON and Bicep templates

Audience profile:

Software engineers tasked with authoring cloud-native solutions that leverage Azure Cosmos DB SQL API and its various SDKs. They are familiar with C#, Python, Java, or JavaScript. They also have experience writing code that interacts with a SQL or NoSQL database platform.

CONSPECT:

- **Get started with Azure Cosmos DB SQL API**
 - Introduction to Azure Cosmos DB SQL API
 - Try Azure Cosmos DB SQL API
 - Lab : Exercise: Create an Azure Cosmos DB SQL API account
- **Plan and implement Azure Cosmos DB SQL API**
 - Plan Resource Requirements
 - Configure Azure Cosmos DB SQL API database and containers
 - Moving data into and out of Azure Cosmos DB SQL API
 - Lab : Exercise: Configure throughput for Azure Cosmos DB SQL API with the Azure portal
 - Lab : Exercise: Migrate existing data using Azure Data Factory

- **Connect to Azure Cosmos DB SQL API with the SDK**
 - Use the Azure Cosmos DB SQL API SDK
 - Configure the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Configure the Azure Cosmos DB SQL API SDK for offline development
 - Lab : Exercise: Connect to Azure Cosmos DB SQL API with the SDK
- **Access and manage data with the Azure Cosmos DB SQL API SDKs**
 - Implement Azure Cosmos DB SQL API point operations
 - Perform cross-document transactional operations with the Azure Cosmos DB SQL API
 - Process bulk data in Azure Cosmos DB SQL API
 - Lab : Exercise: Create and update documents with the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Batch multiple point operations together with the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Move multiple documents in bulk with the Azure Cosmos DB SQL API SDK
- **Execute queries in Azure Cosmos DB SQL API**
 - Query the Azure Cosmos DB SQL API
 - Author complex queries with the Azure Cosmos DB SQL API
 - Lab : Exercise: Paginate cross-product query results with the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Execute a query with the Azure Cosmos DB SQL API SDK
- **Define and implement an indexing strategy for Azure Cosmos DB SQL API**
 - Define indexes in Azure Cosmos DB SQL API
 - Customize indexes in Azure Cosmos DB SQL API
 - Lab : Exercise: Review the default index policy for an Azure Cosmos DB SQL API container with the portal
 - Lab : Exercise: Configure an Azure Cosmos DB SQL API container's index policy with the portal
- **Integrate Azure Cosmos DB SQL API with Azure services**
 - Consume an Azure Cosmos DB SQL API change feed using the SDK
 - Handle events with Azure Functions and Azure Cosmos DB SQL API change feed
 - Search Azure Cosmos DB SQL API data with Azure Cognitive Search
 - Lab : Exercise: Archive Azure Cosmos DB SQL API data using Azure Functions
 - Lab : Exercise: Process change feed events using the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Archive data using Azure Functions and Azure Cosmos DB SQL API
- **Implement a data modeling and partitioning strategy for Azure Cosmos DB SQL API**
 - Model and partition your data in Azure Cosmos DB
 - Optimize databases by using advanced modeling patterns for Azure Cosmos DB
 - Lab : Exercise: Measure performance for customer entities

- Lab : Exercise: Advanced modeling patterns
- **Design and implement a replication strategy for Azure Cosmos DB SQL API**
 - Configure replication and manage failovers in Azure Cosmos DB
 - Use consistency models in Azure Cosmos DB SQL API
 - Configure multi-region write in Azure Cosmos DB SQL API
 - Lab : Exercise: Configure consistency models in the portal and the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Connect to different regions with the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Connect to a multi-region write account with the Azure Cosmos DB SQL API SDK
- **Optimize query performance in Azure Cosmos DB SQL API**
 - Choosing indexes in Azure Cosmos DB SQL API
 - Optimize queries in Azure Cosmos DB SQL API
 - Implement integrated cache
 - Lab : Exercise: Optimize an Azure Cosmos DB SQL API container's index policy for common operations
 - Lab : Exercise: Optimize an Azure Cosmos DB SQL API container's index policy for a specific query
- **Administrating and Monitoring tasks for an Azure Cosmos DB SQL API solution**
 - Measure performance in Azure Cosmos DB SQL API
 - Monitor responses and events in Azure Cosmos DB SQL API
 - Implementing backup and restore for Azure Cosmos DB SQL API
 - Implement security in Azure Cosmos DB SQL API
 - Lab : Exercise: Troubleshoot an application using the Azure Cosmos DB SQL API SDK
 - Lab : Exercise: Use Azure Monitor to analyze an Azure Cosmos DB SQL API account
 - Lab : Exercise: Recover a database or container from a recovery point
 - Lab : Exercise: Store Azure Cosmos DB SQL API account keys in Azure Key Vault
- **Manage an Azure Cosmos DB SQL API solution using DevOps practices**
 - Write scripts for Azure Cosmos DB SQL API
 - Create resource template for Azure Cosmos DB SQL API
 - Lab : Exercise: Adjust provisioned throughput using an Azure CLI script
 - Lab : Exercise: Create an Azure Cosmos DB SQL API container using Azure Resource Manager templates
- **Create server-side programming constructs in Azure Cosmos DB SQL API**
 - Build multi-item transactions with the Azure Cosmos DB SQL API
 - Expand query and transaction functionality in Azure Cosmos DB SQL API
 - Lab : Exercise: Implement and then use a UDF using the SDK

- Lab : Exercise: Create a stored procedure with the Azure Portal

REQUIREMENTS:

Before attending this course, students must have:

- Knowledge of Microsoft Azure and ability to navigate the Azure portal (AZ-900 equivalent)
- Experience writing in an Azure-supported language at the intermediate level. (C#, JavaScript, Python, or Java)

Ability to write code to connect and perform operations on a SQL or NoSQL database product. (SQL Server, Oracle, MongoDB, Cassandra or similar)

Difficulty level



CERTIFICATE:

Certificate of completing an authorized Microsoft training

TRAINER:

Microsoft Certified Trainer