DP-420: Azure Cosmos DB Developer Specialty

Content

Module 1: Get Started with Azure Cosmos DB SQL API

- **Lessons**
  - Introduction to Azure Cosmos DB SQL API
  - Try Azure Cosmos DB SQL API
- **Lab**: Exercise: Create an Azure Cosmos DB SQL API account
- **After completing this module, students will be able to**:
  - Evaluate whether Azure Cosmos DB SQL API is the right database for your application
  - Describe how the features of the Azure Cosmos DB SQL API are appropriate for modern applications
  - Create a new Azure Cosmos DB SQL API account
  - Create database, container, and item resources for an Azure Cosmos DB SQL API account

Module 2: Plan and Implement Azure Cosmos DB SQL API

- **Lessons**
  - 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
- **After completing this module, students will be able to**:
  - Evaluate various requirements of your application
  - Plan for scale and retention requirements
  - Configure throughput allocation
  - Configure time-to-live values
  - Migrate data using Azure services
  - Migrate data using Spark or Kafka

Module 3: Connect to Azure Cosmos DB SQL API with the SDK

- **Lessons**
  - 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
- **After completing this module, students will be able to**:
  - Integrate the Microsoft.Azure.Cosmos SDK library from NuGet
  - Connect to an Azure Cosmos DB SQL API account using the SDK and .NET

- o Configure the SDK for offline development
- o Troubleshoot common connection errors
- o Implement parallelism in the SDK
- o Configure logging using the SDK

Module 4: Access and Manage Data with the Azure Cosmos DB SQL API SDKs

- **Lessons**
  - o Implement Azure Cosmos DB SQL API point operations
  - o Perform cross-document transactional operations with the Azure Cosmos DB SQL API
  - o 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
- **After completing this module, students will be able to**:
  - o Perform CRUD operations using the SDK
  - o Configure TTL for a specific document
  - o Implement optimistic concurrency control for an operation
  - o Create a transactional batch and review results
  - o Create a bulk operation
  - o Review the results of a bulk operation
  - o Implement bulk operation best practices

Module 5: Execute Queries in Azure Cosmos DB SQL API

- **Lessons**
  - o Query the Azure Cosmos DB SQL API
  - o 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
- **After completing this module, students will be able to**:
  - o Create and execute a SQL query
  - o Project query results
  - o Use built-in functions in a query
  - o Implement a correlated subquery
  - o Create a cross-product query

Module 6: Define and Implement an Indexing Strategy for Azure Cosmos DB SQL API

- **Lessons**
  - o Define indexes in Azure Cosmos DB SQL API
  - o 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
- **After completing this module, students will be able to**:
    - View and understand the default indexing policy for a SQL API container
    - Customize the indexing policy for a container
    - Use a composite index in an indexing policy

Module 7: Integrate Azure Cosmos DB SQL API with Azure Services

- **Lessons**
    - 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
- **After completing this module, students will be able to**:
    - Process change feed events using the SDK
    - Implement change feed best practices
    - Create an Azure Functions trigger for Azure Cosmos DB
    - Create an Azure Functions input for Azure Cosmos DB
    - Index Azure Cosmos DB data in Azure Cognitive Search

Module 8: Implement a Data Modeling and Partitioning Strategy for Azure Cosmos DB SQL API

- **Lessons**
    - 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
- **After completing this module, students will be able to**:
    - Identify application access patterns for an existing application
    - Decide when to embed or reference data
    - Use change feed to manage referential integrity
    - Combine multiple entities in a single container
    - Denormalize aggregated data in a single container

Module 9: Design and Implement a Replication Strategy for Azure Cosmos DB SQL API

- **Lessons**
    - 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
- **After completing this module, students will be able to**:
    - Distribute data across various geographies
    - Define automatic failover policies
    - Perform manual failovers
    - Configure default consistency model
    - Change per-session consistency model
    - Configure multi-region write in the SDK
    - Create a custom conflict resolution policy

Module 10: Optimize Query Performance in Azure Cosmos DB SQL API

- **Lessons**
    - 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
- **After completing this module, students will be able to**:
    - Review and compare read-heavy vs. write-heavy index patterns
    - Update indexing policy to optimize index performance
    - Measure cost of a query in request units (RUs)
    - Measure cost of point operations
    - Work with item and query integrated cache
    - Configure integrated cache staleness

Module 11: Administrate and Monitor Tasks for an Azure Cosmos DB SQL API Solution

- **Lessons**
    - 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
- **After completing this module, students will be able to**:
    - Observe rate-limiting events in a container or database
    - Query resource logs using Azure Monitor
    - Review and observe transient and rate-limiting errors
    - Configure alerts

- o Configure continuous backup and recovery
- o Perform a point-in-time recovery
- o Use role-based access control (RBAC)
- o Access account resources using Azure AD and Microsoft Identity Platform

Module 12: Manage an Azure Cosmos DB SQL API Solution Using DevOps Practices

- **Lessons**
  - o Write scripts for Azure Cosmos DB SQL API
  - o 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
- **After completing this module, students will be able to**:
  - o View arguments, groups, and subgroups for a specific CLI command
  - o Create Azure Cosmos DB accounts, databases, and containers using the CLI
  - o Manage an indexing policy using the CLI
  - o Configure database or container throughput using the CLI
  - o Initiate failovers and manage failover regions using the CLI
  - o Review ARM templates
  - o Deploy Azure Cosmos DB SQL API using ARM templates