Day 1

- Introduction to Azure Cosmos DB
    - What Is Azure Cosmos DB?
    - How does Azure Cosmos DB for NoSQL work?
    - When should you use Azure Cosmos DB for NoSQL?
    - Major Features
    - Turnkey Global Distribution
    - Multiple Data Models and APIs
    - Elastically Scale Throughput and Storage on Demand
    - High Availability and Response Time
    - Five Consistency Models
    - Exercise: Setting Up the Development Environment
    - Exercise: Installing Microsoft Visual Studio
    - Exercise: Installing the Azure Cosmos DB Emulator
- Learning Azure Cosmos DB Concepts
    - Understanding Global Distribution
    - Introducing Write and Read Regions
    - Understanding the Consistency Models
    - Scope of Consistency
    - Strong Consistency Model
    - Eventual Consistency Model
    - Bounded Staleness Consistency Model
    - Session Consistency Model
    - Consistent Prefix Consistency Model

    - Consistency for Queries
    - Understanding Partitioning
    - What Are Containers?
    - How Does Partitioning Work?
    - Designing for Partitioning
    - Understanding Throughput
    - Specifying Request Unit Capacity
    - Estimating Throughput
    - Implementing Security
    - Encryption at Rest
    - Firewall Support
    - Securing Access to Data
    - Supported APIs
    - Azure Cosmos DB REST API
    - DocumentDB API
    - MongoDB API
    - Graph API
    - Table API
- Plan Resource Requirements
    - Understand throughput
    - Evaluate throughput requirements
    - Evaluate data storage requirements
    - Time-to-live (TTL)
    - Plan for data retention with time-to-live (TTL)

Day 2

- Configure Azure Cosmos DB for NoSQL database and containers
    - Serverless
    - Compare serverless vs. provisioned throughput
    - Autoscale throughput
    - Compare autoscale vs. standard (manual) throughput
    - Migrate between standard (manual) and autoscale throughput
    - Exercise: Configure throughput for Azure Cosmos DB SQL API with the Azure portal
- Move data into and out of Azure Cosmos DB for NoSQL
    - Move data by using Azure Data Factory
    - Exercise: Migrate existing data using Azure Data Factory
- Use the Azure Cosmos DB for NoSQL SDK
    - Understand the SDK
    - Import from package manager
    - Connect to an online account
    - Implement client singleton
    - Configure connectivity mode
    - Exercise: Connect to Azure Cosmos DB for NoSQL with the SDK
- Configure the Azure Cosmos DB for NoSQL SDK
    - Enable offline development
    - Handle connection errors
    - Implement threading and parallelism
    - Configure logging
    - Exercise: Configure the Azure Cosmos DB for NoSQL SDK for offline development

- Implement Azure Cosmos DB for NoSQL point operations
    - Understand point operations
    - Create documents
    - Read a document
    - Update documents
    - Configure time-to-live (TTL) value for a specific document
    - Delete documents
    - Exercise: Create and update documents with the Azure Cosmos DB for NoSQL SDK
- Perform cross-document transactional operations with the Azure Cosmos DB for NoSQL
    - Create a transactional batch with the SDK
    - Review batch operation results with the SDK
    - Exercise: Batch multiple point operations together with the Azure Cosmos DB for NoSQL SDK
    - Implement optimistic concurrency control
- Process bulk data in Azure Cosmos DB for NoSQL
    - Introduction
    - Create bulk operations with the SDK
    - Review bulk operation caveats
    - Implement bulk best practices
    - Exercise: Move multiple documents in bulk with the Azure Cosmos DB for NoSQL SDK

Day 3

- Query the Azure Cosmos DB for NoSQL
    - Understand SQL query language
    - Create queries with SQL
    - Project query results
    - Implement type-checking in queries
    - Use built-in functions
    - Execute queries in the SDK
    - Exercise: Execute a query with the Azure Cosmos DB for NoSQL SDK
- Author complex queries with the Azure Cosmos DB for NoSQL
    - Create cross-product queries
    - Implement correlated subqueries
    - Implement variables in queries
    - Paginate query results
    - Exercise: Paginate cross-product query results with the Azure Cosmos DB for NoSQL SDK
- Define indexes in Azure Cosmos DB for NoSQL
    - Introduction
    - Understand indexes
    - Understand indexing policies
    - Review indexing policy strategies
    - Exercise: Review the default index policy for an Azure Cosmos DB for NoSQL container with the portal

- Customize indexes in Azure Cosmos DB for NoSQL
    - Customize the indexing policy
    - Evaluate composite indexes
    - Exercise: Configure an Azure Cosmos DB for NoSQL container's index policy with the portal
- Consume an Azure Cosmos DB for NoSQL change feed using the SDK
    - Understand change feed features in the SDK
    - Implement a delegate for the change feed processor
    - Implement the change feed processor
    - Implement the change feed estimator
    - Exercise: Process change feed events using the Azure Cosmos DB for NoSQL SDK
- Handle events with Azure Functions and Azure Cosmos DB for NoSQL change feed
    - Understand Azure Function bindings for Azure Cosmos DB for NoSQL
    - Configure function bindings
    - Develop function
    - Exercise: Process Azure Cosmos DB for NoSQL data using Azure Functions
- Search Azure Cosmos DB for NoSQL data with Azure Cognitive Search
    - Create an indexer for data in Azure Cosmos DB for NoSQL
    - Implement a change detection policy
    - Manage a data deletion detection policy
    - Exercise - Search data using Azure Cognitive Search and Azure Cosmos DB for NoSQL

Day 4

- Implement a non-relational data model
    - What's the difference between NoSQL and relational databases?
    - Identify access patterns for your app
    - When to embed or reference data
    - Exercise: Measure performance for customer entities
    - Choose a partition key
    - Model small lookup entities
- Design a data partitioning strategy
    - Introduction
    - Denormalize data in your mode
    - Manage referential integrity by using change feed
    - Combine multiple entities in the same container
    - Denormalize aggregates in the same container
    - Finalize the data model
    - Exercise advanced modeling patterns
- Configure replication and manage failovers in Azure Cosmos DB
    - Introduction
    - Understand replication
    - Distribute data across regions
    - Evaluate the cost of distributing data globally
    - Define automatic failover policies
    - Perform manual failovers

    - Configure SDK region
    - Exercise: Connect different regions with the Azure Cosmos DB for NoSQL SDK
- Use consistency models in Azure Cosmos DB for NoSQL
    - Configure default consistency model in the portal
    - Change consistency model with the SDK
    - Use session tokens
    - Exercise: Configure consistency models in the portal and the Azure Cosmos DB for NoSQL SDK
- Configure multi-region write in Azure Cosmos DB for NoSQL
    - Understand multi-region write
    - Configure multi-region support in the SDK
    - Understand conflict resolution policies
    - Create custom conflict resolution policy
    - Exercise: Connect multi-region write account with the Azure Cosmos DB for NoSQL SDK
- Customize an indexing policy in Azure Cosmos DB for NoSQL
    - Index usage
    - Review read-heavy index patterns
    - Review write-heavy index patterns
    - Exercise - Optimize an Azure Cosmos DB for NoSQL container's index policy for common operations

Day 5

- Measure index performance in Azure Cosmos DB for NoSQL
    - Enable indexing metrics
    - Analyze indexing metrics results
    - Measure query cost
    - Measure point operation cost
    - Exercise - Optimize an Azure Cosmos DB for NoSQL container's index policy for a specific query
- Implement integrated cache in Azure Cosmos DB for NoSQL
    - Review workloads that benefit from the cache
    - Enable integrated cache
    - Configure cache staleness
- Measure performance in Azure Cosmos DB for NoSQL
    - Understand Azure Monitor
    - Measure throughput
    - Observe rate-limiting events
    - Query logs
    - Exercise: Use Azure Monitor to analyze an Azure Cosmos DB for NoSQL account
- Implement backup and restore for Azure Cosmos DB for NoSQL
    - Evaluate periodic backup
    - Configure continuous backup and recovery
    - Perform a point-in-time recovery
    - Exercise: Recover a database or container from a recovery point

- 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 Azure Active Directory
    - Understand Always Encrypted
    - Exercise: Store Azure Cosmos DB for NoSQL account keys in Azure Key Vault
- Expand query and transaction functionality in Azure Cosmos DB for NoSQL
    - Create user-defined functions
    - Create user-defined functions with the SDK
    - Add triggers to an operation
    - Create and use triggers with the SDK
    - Exercise: Implement and use user defined functions with the Azure Cosmos DB SDK
- Working with a Globally Distributed Database
    - Configuring Global Distribution
    - Configuring Failover
    - Working with Manual Failover
    - Configuring Automatic Failover
    - Connecting to a Preferred Region
    - Implementing a Multi-Master Database
    - Application Scenario
    - Implementing the Solution