

DATA ENGINEERING AND ANALYTICS

Working with NoSQL Databases (MongoDB and Cosmos DB)

Level: Practitioner • 2 days (expandable to 3) • Virtual, In-person

Overview

NoSQL databases were oversold as the replacement for relational databases, and the industry spent a decade cleaning up after that pitch. The truth is more useful: document databases like MongoDB and globally distributed services like Azure Cosmos DB are excellent tools for specific shapes of data and specific scale requirements, and painful ones when chosen for fashion. The genuinely hard part is not the syntax; it is the modeling. Document databases reward developers who design for how data is read, and punish those who bring relational habits (or no habits) to a schema-flexible world.

This is a hands-on, practitioner course. It starts with the judgment question of when NoSQL is the right choice, because that decision shapes everything after it. From there it builds skills in order: document data modeling, then working fluently with MongoDB, then indexing and performance, then Azure Cosmos DB with its partitioning and consistency model, and finally the operational judgment of running these systems for real. In keeping with a less-but-deeper philosophy, the course goes deep on the document model, the one most teams actually adopt, rather than surveying every NoSQL category at equal depth. Every module ends with a lab, and each module builds on the one before.

Who Should Attend

- Developers building applications on MongoDB or Cosmos DB, or about to
- Database professionals extending relational experience into non-relational systems
- Architects deciding where NoSQL belongs (and does not) in their data landscape

Prerequisites

- Working experience with a relational database and SQL
- Comfort with JSON and basic programming in any language
- Familiarity with relational modeling helps; *Relational Database Design* is the companion course on that side

What You Will Learn

- Judge when a document database is the right choice, and when relational remains better
- Design document models around access patterns: embedding versus referencing, done deliberately
- Query MongoDB fluently, including the aggregation pipeline
- Index for real workloads and diagnose slow queries
- Apply Cosmos DB's core concepts: partition keys, request units, and consistency levels
- Operate NoSQL systems sensibly: schema versioning, migration, and cost control

Course Outline

Day one: the document model and MongoDB fluency

- When NoSQL Is the Right Answer
 - The NoSQL landscape in one map: document, key-value, column-family, and graph
 - What you trade away: joins, transactions, and schema enforcement, and what you get back
 - Consistency tradeoffs in plain language: why distributed databases make you choose
 - Lab: evaluate three realistic scenarios and defend a relational or NoSQL choice for each
- Modeling Documents
 - Designing for access patterns: the question that replaces normalization
 - Embedding versus referencing, and the honest costs of each
 - Handling relationships, arrays, and data that grows without bound
 - Lab: design document models for a domain you modeled relationally before, and compare
- Working with MongoDB
 - CRUD operations and query operators that cover daily work
 - The aggregation pipeline: filtering, grouping, and reshaping documents
 - Lab: implement the day's data model in MongoDB and answer business questions with aggregations

Day two: performance, Cosmos DB, and operating for real

- Indexing and Performance in MongoDB
 - How MongoDB indexes work, and the compound index rules worth memorizing
 - Reading query plans and finding the slow query's real cause
 - Write concerns, read preferences, and what replication means for your data
 - Lab: diagnose and fix a set of slow queries with indexes and model adjustments
- Azure Cosmos DB
 - What Cosmos DB is: a globally distributed service with multiple APIs, including MongoDB compatibility
 - Partition keys: the one decision you cannot easily undo
 - Request units, throughput provisioning, and the five consistency levels in practice
 - Lab: build a container with a well-chosen partition key and measure request unit costs of queries
- Operating NoSQL in Production
 - Schema flexibility versus schema chaos: versioning documents over time
 - Migration and coexistence: NoSQL alongside your relational systems
 - Cost, backup, and monitoring habits that prevent unpleasant surprises
 - Lab: version a document schema and migrate existing data without downtime

Extended Version

The three-day version keeps the same gradient and adds depth and a fuller build:

- Multi-document transactions and change streams in MongoDB
- Global distribution in Cosmos DB: multi-region writes and conflict handling
- A deeper look at modeling hard cases: many-to-many at scale and time-series data

- A capstone that designs, builds, and load-tests a document-backed service end to end, with partitioning and indexing defened