

MS-DP800T00: DEVELOP AI-ENABLED DATABASE SOLUTIONS



DURATION	LEVEL	TECHNOLOGY	DELIVERY METHOD	TRAINING CREDITS
3 Days	Intermediate	Azure / SQL Database	Instructor-led / VILT	N/A

INTRODUCTION

This course provides students with the knowledge and skills to design and develop AI enabled database solutions across Microsoft SQL platforms, including SQL Server, Azure SQL, and SQL databases in Microsoft Fabric. It is intended for professionals who build modern data solutions that integrate structured and semi structured data and incorporate AI features into scalable enterprise applications. It will also be valuable for individuals who develop applications that rely on SQL based data services enhanced with vector search, embeddings, and other AI driven capabilities.

AUDIENCE PROFILE

The audience for this course is data professionals who want to learn about designing and developing AI-enabled database solutions across Microsoft's SQL platforms, including SQL Server, Azure SQL, and SQL databases in Microsoft Fabric. This role develops database solutions that include both structured and semi-structured data and integrates AI features into modern and highly scalable enterprise applications.

PREREQUISITES

- Experience writing T-SQL queries.
- Basic understanding of database concepts such as tables, joins, and transactions
- Familiarity with SQL Server or Azure SQL Databases in Microsoft Fabric.
- Basic understanding of security concepts like authentication and authorization.
- Experience with Git version control and CI/CD concepts.

COURSE OBJECTIVES

After completing this course, students will be able to:

- Create well-structured database objects and indexes
- Encapsulate business logic with stored procedures and functions
- Write advanced T-SQL using techniques such as Common Table Expressions (CTE), window functions, and error handling.
- Accelerate your development workflow with AI-assisted tools including GitHub Copilot and Fabric Copilot.
- Take your database solutions from development to production.
- Protect sensitive data with encryption, masking, and row-level security.
- Automate deployments with CI/CD pipelines using SQL Database Projects

COURSE CONTENT

Module 1: Design and implement database objects with SQL

This module covers designing and implementing various database objects including tables with different data types, specialized table types, indexes, constraints, and partitioning strategies. You'll learn how to create and optimize database objects for modern SQL platforms.

Lessons:

- Introduction
- Understand your SQL Server-based platform choices

- Build effective tables
- Optimize with indexes
- Use specialized table types
- Enforce data integrity with constraints
- Manage JSON columns and indexes
- Partition tables for scale
- Exercise - Create and maintain database objects
- Module assessment
- Summary

After completing this module, students will be able to:

- Design and implement tables with appropriate data types, sizes, and columns
- Implement specialized tables including in-memory, temporal, external, LEDGER, and GRAPH tables
- Design and implement JSON columns and indexes
- Design and implement database constraints including primary key, foreign key, unique key, CHECK, and DEFAULT

- Design and implement SEQUENCES
- Design and implement partitioning strategies for tables and indexes

Module 2: Implement programmability objects with SQL

Learn how to create and use views, stored procedures, scalar functions, table-valued functions, and triggers to build maintainable, secure, and efficient database solutions.

Lessons:

- Introduction
- Create views
- Create stored procedures
- Create scalar functions
- Create table-valued functions
- Create triggers
- Choose when to use each option
- Exercise: Implement programmability objects in SQL Server
- Knowledge check
- Summary

After completing this module, students will be able to:

- Create views to simplify data access and provide security boundaries.
- Build stored procedures to encapsulate complex business logic and data modifications.
- Develop scalar functions to return single calculated values.
- Implement table-valued functions to return result sets for use in queries.
- Configure triggers to automatically respond to data modifications or database events.
- Choose the right programmability object based on specific requirements.

Module 3: Write advanced T-SQL code

Learn advanced T-SQL techniques including CTEs, window functions, JSON, regular expressions, fuzzy matching, graph queries, and error handling for SQL Server, Azure SQL, and Fabric.

Lessons:

- Introduction
- Organize queries with Common Table Expressions
- Apply window functions for analytics
- Process JSON data with built-in functions
- Match patterns with regular expressions
- Find approximate matches with fuzzy string functions
- Traverse relationships with graph queries

- Compare rows with correlated subqueries
- Handle errors with TRY...CATCH
- Exercise - Work with JSON functions
- Module assessment
- Summary

After completing this module, students will be able to:

- Write Common Table Expressions (CTEs) including recursive CTEs for hierarchical data
- Use window functions for ranking, aggregation, and analytical calculations
- Parse, construct, and transform JSON data using JSON functions
- Apply regular expressions for pattern matching and text manipulation
- Find approximate matches using fuzzy string matching functions
- Create and query graph data using the MATCH operator
- Write correlated subqueries for row-by-row comparisons
- Implement structured error handling with TRY...CATCH

Module 4: Implement SQL solutions by using AI-assisted tools

Learn how to leverage GitHub Copilot and Fabric Copilot for AI-assisted database development across SQL Server, Azure SQL, and SQL databases in Microsoft Fabric.

Lessons:

- Introduction
- Describe AI-assisted development tools available for Microsoft SQL platforms
- Interpret security impact of using AI-assisted tools
- Enable GitHub Copilot and Fabric Copilot
- Configure model and Model Context Protocol (MCP) tool options in a GitHub Copilot or Fabric Copilot chat session
- Create and configure GitHub Copilot instruction files
- Connect to MCP server endpoints, including Microsoft SQL Server and Fabric Lakehouse
- Exercise - Configure AI-assisted tools for database development
- Module assessment
- Summary

After completing this module, students will be able to:

- Describe AI-assisted development tools available for Microsoft SQL platforms
- Interpret security impact of using AI-assisted tools

- Enable GitHub Copilot and Fabric Copilot
- Configure model and Model Context Protocol (MCP) tool options in a GitHub Copilot or Fabric Copilot chat session
- Create and configure GitHub Copilot instruction files
- Connect to MCP server endpoints, including Microsoft SQL Server and Fabric Lakehouse

Module 5: Implement data security and compliance with SQL

Learn how to protect sensitive data and meet compliance requirements by implementing encryption, masking, access controls, and auditing across Microsoft's SQL platforms.

Lessons:

- Introduction
- Protect data with encryption
- Configure dynamic data masking
- Implement row-level security
- Manage permissions and secure access
- Implement auditing
- Configure secure access to AI services
- Secure data API endpoints
- Exercise: Implement security features
- Module assessment
- Summary

After completing this module, you will be able to:

- Design and implement data encryption using Always Encrypted and column-level encryption
- Configure Dynamic Data Masking to protect sensitive data
- Implement Row-Level Security to filter data access based on user context
- Design object-level permissions and implement secure, passwordless database access using Microsoft Entra ID
- Configure auditing to track database activity
- Secure model endpoints and API endpoints for modern applications

Module 6: Optimize database performance

Optimize Azure SQL Database performance by choosing the right service tier and managing concurrency with transaction isolation levels. Analyze queries with execution plans and DMVs. Use Query Store for plan management and diagnose blocking and deadlocks.

Lessons:

- Introduction
- Recommend database configurations
- Preserve data integrity with transaction isolation levels and concurrency controls
- Evaluate query performance with execution plans and DMVs
- Monitor and tune queries with Query Store and Query Performance Insight
- Identify and resolve blocking and deadlocks
- Exercise: Optimize query performance
- Knowledge check
- Summary

After completing this module, you will be able to:

- Evaluate and recommend database configurations including service tiers, compute tiers, and resource limits.
- Choose transaction isolation levels and concurrency controls that balance consistency with throughput.
- Analyze query performance using execution plans and dynamic management views.
- Monitor and tune queries with Query Store and Query Performance Insight.
- Identify and resolve blocking and deadlocks using DMVs and Extended Events.

Module 7: Implement CI/CD by using SQL Database Projects

Implement CI/CD for SQL Database Projects with source control, branching, schema drift detection, automated pipelines, and testing strategies using GitHub Actions and Azure DevOps.

Lessons

- Introduction
- Create, build, and validate SQL Database Projects
- Configure source control and manage reference data
- Manage branching, pull requests, and conflict resolution
- Detect and resolve schema drift
- Implement CI/CD pipelines
- Design and implement a testing strategy
- Exercise: Implement CI/CD by using SQL Database Projects
- Knowledge check
- Summary

After completing this module, you will be able to:

- Create, build, and validate database models by using SQL Database Projects, including SDK-style.

- Configure source control for SQL Database Projects and manage reference data with predeployment and post-deployment scripts.
- Manage branching, pull requests, and conflict resolution for database code.
- Detect schema drift by using schema comparison tools and SqlPackage.
- Implement CI/CD pipelines with GitHub Actions and Azure DevOps, including secrets management and deployment controls.
- Design and implement a testing strategy with unit tests and integration tests.

Module 8: Integrate SQL solutions with Azure services

Create REST and GraphQL APIs for SQL databases using Data API Builder, deploy to Azure hosting services, and implement monitoring and event-driven change patterns.

Lessons

- Introduction
- Create configuration files for Data API Builder
- Define entities for REST and GraphQL
- Expose database objects, stored procedures, and views
- Explore deployment options for Data API Builder
- Recommend Azure Monitor configurations
- Handle changes with event-driven patterns
- Exercise - Configure Data API Builder for a product catalog
- Module assessment
- Summary

After completing this module, you will be able to:

- Create configuration files for Data API Builder with database connections and runtime settings
- Define entities for REST and GraphQL with field mappings, caching, and relationships
- Configure REST and GraphQL endpoints with custom paths and operations
- Expose database views, stored procedures, and GraphQL relationships
- Explore deployment options for Data API Builder, including Azure Container Apps, App Service, and Static Web Apps
- Set up Azure Monitor configurations with Application Insights and Log Analytics
- Handle database changes using Change Data Capture, Azure Functions, and Change Event Streaming

Module 9: Create a knowledge mining solution with Azure AI Search

Unlock the hidden insights in your data with Azure AI Search. In this module, you'll learn how to implement a knowledge mining solution that extracts and enriches data, making it searchable and ready for deeper analysis.

Lessons

- Introduction
- What is Azure AI Search?
- Extract data with an indexer
- Enrich extracted data with AI skills
- Search an index
- Persist extracted information in a knowledge store
- Exercise - Create a knowledge mining solution
- Module assessment
- Summary

After completing this module, you will be able to:

- Implement indexing with Azure AI Search
- Use AI skills to enrich data in an index
- Search an index to find relevant information
- Persist extracted information in a knowledge store

Module 10: Design and implement models and embeddings with SQL

Integrate AI models with Azure SQL Database using external models and built-in AI functions. Design effective embedding strategies and implement maintenance patterns to keep embeddings aligned with source data.

Lessons

- Introduction
- Understand and evaluate models for SQL database workloads
- Create and manage external models in SQL
- Design embeddings for SQL database workloads
- Generate and maintain embeddings for SQL database workloads
- Exercise - Generate and update embeddings in Azure SQL Database
- Knowledge check
- Summary

After completing this module, you will be able to:

- Evaluate AI models for SQL database workloads based on capabilities and performance requirements.
- Create and manage external models to reference AI endpoints from Transact-SQL.
- Design embeddings with appropriate chunking strategies.

- Generate and store embeddings using built-in SQL AI functions.
- Choose maintenance approaches to keep embeddings aligned with source data.

Module 11: Design and implement intelligent search with SQL

Implement intelligent search capabilities in SQL Server and Azure SQL by combining traditional full-text search with semantic vector search. Establish a mental model for different search approaches, prepare SQL for vector-based search, and implement vector, hybrid, and ranking-based search patterns with performance considerations.

Lessons

- Introduction
- Choose an intelligent search approach
- Implement full-text search
- Prepare SQL for vector search
- Implement vector search query patterns
- Implement hybrid search and ranking

- Exercise - Implement intelligent search with full-text, vector, and hybrid queries
- Knowledge check
- Summary

After completing this module, you will be able to:

- Choose between full-text, semantic vector, and hybrid search approaches.
- Implement full-text search for keyword-based queries.
- Design for vector data including vector data type, indexes, and size.
- Evaluate vector index types and metrics and choose between ANN and ENN.
- Implement vector search using vector-related functions.
- Implement hybrid search and Reciprocal Rank Fusion (RRF).
- Evaluate performance of vector and hybrid search.

Module 12: Design and implement RAG with SQL

This module teaches you how to implement Retrieval Augmented Generation (RAG) using Azure SQL Database. You learn to identify

appropriate RAG scenarios, prepare SQL results as LLM context, construct augmented prompts, and process model responses.

Lessons

- Introduction
- Identify RAG use cases and architecture
- Prepare retrieval context for augmentation
- Augment prompts with database context
- Generate and process RAG responses
- Exercise: Implement a RAG solution
- Knowledge check
- Summary

After completing this module, you will be able to:

- Identify use cases for RAG in SQL-based applications
- Convert structured database data to JSON for LLM processing
- Create prompts that combine instructions with retrieved database context
- Send retrieval results to an LLM and extract responses

ASSOCIATED CERTIFICATIONS & EXAM

This course will prepare delegates to write the Microsoft DP-800T00 Exam — Developing AI-Enabled Database Solutions Exam.