

Database Development

Schema design, advanced T-SQL, programmability, and AI-assisted development — 35–40% of DP-800.

This is where you stop writing SQL like a beginner. AIDD-MS-1 takes you from competent query writer to junior database developer — and by the end you've built FieldNote v1.

WHAT YOU'LL BE ABLE TO DO

- **Design normalized schemas** with the right data types, constraints, and indexes.
- **Pick the right table type.** Temporal, in-memory, ledger, graph — and when each is wrong.
- **Write production T-SQL.** Window functions, CTEs, JSON, regex, fuzzy matching, graph.
- **Build programmability.** Views, procedures, functions, triggers — and when not to.
- **Use AI assistants safely.** Copilot, Copilot in Fabric, MCP — with security awareness.

PREREQUISITES

Comfort with SELECT/JOIN/GROUP BY and relational concepts · one query tool (SSMS, ADS, or VS Code) · AIDD-MS-0 or entry-diagnostic pass.

DP-800 COVERAGE

Domain 1: Design and develop database solutions. Weight 35–40%. Every objective has a home here (Appendix A).

BUILD — FIELDNOTE V1

Build the schema, queries, and programmability layer of a real working application — the same one you'll harden in MS-2 and AI-enable in MS-3.

LESSONS	TIME	FORMAT	EXAM
28	~22–28h	Hybrid	Domain 1

Ready to write SQL like a senior?

Cohort dates and enrollment at doctaco.academy/aidd

AIDD-MS-1 · Page 1 of 2

Inside AIDD-MS-1

Module 1A · Designing & Implementing Database Objects

8 LESSONS · ~6 HOURS

Tables, data types, sizing. Rowstore and columnstore indexes. In-memory, temporal, external, ledger, and graph tables. JSON columns and indexes. Constraints, sequences, partitioning.

[Data types](#) · [Indexes](#) · [Temporal](#) · [Graph](#) · [JSON](#) · [Partitioning](#)

Module 1B · Programmability Objects

6 LESSONS · ~4 HOURS

Views and indexed views. Scalar functions and their performance trap. Inline vs multi-statement TVFs. Stored procedures with parameters, output, and error patterns. Triggers — and when not to use them.

[Views](#) · [Functions](#) · [Stored procedures](#) · [Triggers](#) · [Error handling](#)

Module 1C · Advanced T-SQL

9 LESSONS · ~6.5 HOURS

Recursive CTEs. Window functions for ranking, aggregation, and analytics. JSON functions, regex, fuzzy string matching. Graph queries with MATCH. Correlated subqueries and TRY/CATCH.

[CTEs](#) · [Window functions](#) · [JSON_VALUE](#) · [REGEXP_LIKE](#) · [JARO_WINKLER](#) · [MATCH](#)

Module 1D · AI-Assisted SQL Development

5 LESSONS · ~3.5 HOURS

Security when using AI against real data. Configuring GitHub Copilot and Copilot in Fabric. Model and MCP tool selection. Copilot instruction files. Connecting to MCP server endpoints.

[GitHub Copilot](#) · [Copilot in Fabric](#) · [MCP](#) · [Instruction files](#) · [AI security](#)

Module 1E · Mini-Capstone — Build FieldNote v1

8-12 HOURS

Design and implement the data layer of a real SaaS app: temporal tables, JSON, graph relationships; full constraints and indexing; procedures, functions, views; AI integration via Copilot instruction files and a DAB+MCP configuration. Reviewed in a 60-minute reflection session.