

## DATA ENGINEERING AND ANALYTICS

# SQL Querying and T-SQL Fundamentals

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

## Overview

SQL is fifty years old and still the single most useful data skill there is: analysts, developers, testers, and managers who can query a database answer their own questions instead of waiting in someone else's queue. It is also deceptively easy to learn badly. Anyone can copy a SELECT statement; the durable skill is thinking in sets, understanding what a join actually does, and knowing why a query returns the rows it returns, especially when NULLs and grouping get involved.

This is a hands-on, foundation course. It builds SQL in strict dependency order: reading rows from one table, then combining tables with joins, then summarizing with aggregation, then composing larger questions with subqueries and common table expressions, and finally modifying data safely. The dialect is T-SQL on Microsoft SQL Server, and the concepts transfer directly to every other relational database. In keeping with a less-but-deeper philosophy, the course drills a compact set of query patterns until they are second nature rather than touring every language feature. Every module ends with a lab against a realistic sample database, and each module builds on the one before.

## Who Should Attend

- Analysts, testers, and support engineers who need to query databases directly
- Developers who use databases but have never been formally taught SQL
- Anyone starting a data career who needs the one skill every data role assumes

## Prerequisites

- No SQL or database experience required
- Comfort with basic spreadsheet concepts (rows, columns, sorting) is enough
- A laptop able to run the provided lab environment

## What You Will Learn

- Read and write SELECT queries with confident filtering and sorting
- Combine tables with inner and outer joins, and predict what each join returns
- Summarize data with aggregate functions, GROUP BY, and HAVING
- Handle NULLs, data types, and the T-SQL functions used daily for strings and dates
- Compose complex questions with subqueries and common table expressions
- Insert, update, and delete data safely, with transactions as a seatbelt

## Course Outline

### Day one: reading data well

- SELECT Fundamentals
  - Tables, rows, and columns: the mental model behind every query
  - SELECT and WHERE: choosing columns and filtering rows with confidence

- Sorting, TOP, and DISTINCT, and the logical order queries actually run in
- Lab: answer a first set of business questions from a single table
- Joining Tables
  - Why data lives in multiple tables, and how keys connect them
  - INNER JOIN step by step: predicting exactly which rows survive
  - LEFT and RIGHT outer joins, and spotting the classic missing-rows mistakes
  - Lab: answer questions that require joining two and three tables
- Aggregation
  - COUNT, SUM, AVG, MIN, and MAX: from rows to answers
  - GROUP BY: summarizing by category without the errors everyone makes first
  - HAVING versus WHERE: filtering groups versus filtering rows
  - Lab: build a set of summary reports with grouping and aggregate filters

### **Day two: T-SQL essentials and changing data**

- Working with Real-World Data
  - NULL: what it means and how it quietly changes comparisons and counts
  - Data types and conversion in T-SQL
  - The everyday functions: strings, dates, and CASE for conditional logic
  - Lab: clean up and reformat messy query output into a presentable result
- Composing Bigger Questions
  - Subqueries: queries inside queries, and where they belong
  - Common table expressions: making complex logic readable
  - A first look at window functions: ranking and running totals
  - Lab: solve multi-step business questions with CTEs and a window function
- Modifying Data Safely
  - INSERT, UPDATE, and DELETE, and why WHERE matters most here
  - Transactions: trying a change before committing to it
  - Reading an unfamiliar database: schemas, keys, and finding your way around
  - Lab: make a set of data changes inside transactions and verify each one before committing

### **Extended Version**

The three-day version keeps the same gradient and adds practice depth where fluency comes from:

- Deeper window function work: partitions, frames, and practical reporting patterns
- More complex multi-table scenarios and query troubleshooting technique
- An introduction to views, stored procedures, and query performance basics
- A capstone that answers an open-ended business brief from a full sample database, queries reviewed together as a class