

DATABASES

INTERMEDIATE

3 DAYS

T-SQL Programming

Three days of Transact-SQL for developers and BI professionals writing production code against SQL Server. Covers advanced querying and set operations, window functions, data modification including MERGE, transactions and structured error handling, stored procedures, functions, triggers, temporal tables, safe dynamic SQL, and the query patterns that keep code performing as data grows. Delivered on SQL Server 2022 and 2025 with Azure SQL notes.

Why T-SQL Programming Training?

Set-based thinking

Cursors and row-by-row loops get replaced with set-based patterns, which is usually the single largest performance gain available.

Injection-safe by construction

Dynamic SQL is covered with parameterisation and quoting practices rather than avoided as too risky to teach.

Error handling that holds up

TRY CATCH, transaction state and retry logic are taught properly, because half-committed work is expensive to unwind.

Performance built into the writing

Participants read execution plans for their own code, so efficiency becomes a habit rather than a later tuning exercise.

What you'll be able to do

- ✓ Explain logical query processing order and use it to debug unexpected results
- ✓ Write advanced joins, subqueries, set operators and APPLY expressions
- ✓ Apply window functions for ranking, running totals and period comparison
- ✓ Handle dates, strings, types and collations correctly
- ✓ Modify data safely using INSERT, UPDATE, DELETE, MERGE and OUTPUT
- ✓ Control transactions and implement structured error handling with TRY CATCH
- ✓ Build maintainable stored procedures with appropriate parameter handling
- ✓ Choose between scalar, inline and table valued functions on performance grounds
- ✓ Implement triggers, views and temporal tables appropriately
- ✓ Write dynamic SQL that is parameterised and injection resistant
- ✓ Work with JSON and XML data in T-SQL
- ✓ Read execution plans to validate that code performs at scale

Who this is for

- Application developers writing T-SQL against SQL Server
- BI and reporting developers
- Data engineers building SQL Server based pipelines
- Database developers moving from Oracle, PostgreSQL or MySQL
- Analysts progressing from ad hoc querying to production code

Curriculum

01 T-SQL Foundations and Query Processing

- Logical query processing order and why it explains most surprises
- Three valued logic, nulls and predicate behaviour
- Data types, precedence and implicit conversion pitfalls
- Collation, comparison and sorting behaviour
- Batch structure, GO and variable scope
- **Hands-on:** debug a set of queries that return unexpected results

02 Advanced Querying and Set Operations

- Inner, outer, cross and self joins and reading the resulting row counts
- Correlated and non-correlated subqueries
- CROSS APPLY and OUTER APPLY for per-row evaluation
- UNION, INTERSECT and EXCEPT for comparison and reconciliation
- Common table expressions and recursive CTEs
- **Hands-on:** rewrite a nested legacy query into readable, verifiable steps

03 Window Functions and Analytical T-SQL

- OVER, PARTITION BY, ORDER BY and frame specification
- ROW_NUMBER, RANK, DENSE_RANK and NTILE
- LAG, LEAD, FIRST_VALUE and LAST_VALUE
- Running totals, moving averages and cumulative measures
- GROUPING SETS, ROLLUP and CUBE for subtotals
- **Hands-on:** build a period comparison report in a single query

04 Dates, Strings and Data Handling

- Date and time types, offsets and time zone handling
- Date arithmetic, boundaries and fiscal period calculation
- String functions, STRING_SPLIT and STRING_AGG
- Formatting, parsing and safe conversion with TRY_CONVERT
- Handling nulls with ISNULL, COALESCE and NULLIF
- **Hands-on:** clean and standardise a messy inbound dataset

05 Modifying Data Safely

- INSERT variants, SELECT INTO and bulk insert patterns
- UPDATE and DELETE with joins and their common mistakes
- MERGE syntax, use cases and the known caveats
- The OUTPUT clause for capturing changed rows
- Batched modification for very large tables
- **Hands-on:** implement an upsert and a batched delete on a large table

06 Transactions and Error Handling

- Explicit, implicit and autocommit transaction behaviour
- TRY CATCH structure and what it does and does not catch
- XACT_STATE, XACT_ABORT and doomed transactions
- THROW and RAISERROR and designing error contracts
- Savepoints, nested transactions and retry logic
- **Hands-on:** build a procedure that fails safely and retries correctly

07 Stored Procedures

- Procedure structure, parameters and output parameters
- Return values, result sets and calling conventions
- Parameter sniffing, plan reuse and RECOMPILE options
- Table valued parameters for set-based inputs
- Versioning, deployment and procedure coding standards
- **Hands-on:** build a parameterised procedure and stabilise its plan behaviour

08 Functions and Reusable Logic

- Scalar functions and their historic performance cost
- Inline table valued functions as the preferred pattern
- Multi-statement table valued functions and estimation problems
- Scalar UDF inlining and what changed in recent versions
- Deterministic functions, schema binding and indexed views
- **Hands-on:** convert a scalar function to an inline equivalent and compare plans

09 Views, Triggers and Temporal Tables

- Views, nesting depth and predicate pushdown behaviour
- Indexed views and their maintenance cost
- DML triggers, INSTEAD OF triggers and the inserted and deleted tables
- Trigger anti-patterns and multi-row correctness
- System versioned temporal tables for history and point-in-time queries
- **Hands-on:** implement temporal history and query the data as of a past date

10 Dynamic SQL and Secure Coding

- When dynamic SQL is justified and when it is not
- sp_executesql, parameterisation and plan reuse
- QUOTENAME and identifier handling
- SQL injection mechanics and defensive coding
- Execution context, EXECUTE AS and least privilege for procedures
- **Hands-on:** build a dynamic search procedure and prove it resists injection

11 JSON, XML and Writing for Performance

- JSON functions, OPENJSON and JSON path expressions
- FOR JSON and FOR XML output shaping
- XML data type, XQuery and when XML still applies
- SARGable predicates and writing code the optimiser can index
- Reading execution plans for your own procedures
- Code review checklist for production T-SQL
- **Hands-on:** capstone procedure delivered with tests and a plan review