Mastering SQL with SQL Server 2022

Course Description: This comprehensive training course is designed for developers and data professionals aiming to master SQL querying and data manipulation using Microsoft SQL Server 2022. Participants will gain a strong foundation in relational database concepts, T-SQL syntax, and performance-optimized querying techniques. The course emphasizes hands-on practice with real-world datasets, deep dives into multi-table joins, subqueries, window functions, and introduces modern features like JSON, PIVOT, and execution plan analysis. Whether you're building reports, integrating data, or developing applications, this course will equip you with the tools to write efficient, maintainable, and scalable SQL code.

Duration : 40 hours(5 Days)

Target Audience:

- Application Developers
- Data Analysts
- Backend Engineers
- BI Developers using SQL Server
- Anyone looking to write better SQL queries with confidence

Course Content Overview:

Module 1: Introduction and Orientation

- Overview of MS SQL Server 2022 for Developers
- Developer Tools and Query Editors
- Labs :
    - **Lab 1.1** – Connect to SQL Server with SSMS
    - **Lab 1.2** – Explore system databases, create your own practice database

Module 2: SQL and Relational Database Fundamentals

- Understanding Relational Databases and Normalization
- Core Concepts: Tables, Columns, Rows, Keys (Primary, Foreign)
- Data Relationships: 1:1, 1:M, M:M

- Database Schemas and Data Types
- Intro to SQL Server Metadata & Information Schema
- Labs :
    - **Lab 2.1** – Create Tables with Primary and Foreign Keys
    - **Lab 2.2** – Insert Sample Records
    - **Lab 2.3** – Practice normalization using sample data
    - **Lab 2.4** – Use INFORMATION_SCHEMA to inspect database metadata

Module 3: Basic Querying Techniques

- Writing SELECT Queries: Column Selection, Table Aliases
- Filtering Data with WHERE, NULL Logic
- Sorting Data using ORDER BY (ASC/DESC)
- Using DISTINCT and TOP for Result Control
- Column Expressions, Aliases, and Basic Arithmetic
- String Manipulations: LEFT(), RIGHT(), LEN(), CHARINDEX(), REPLACE()
- Date Functions: GETDATE(), DATEPART(), FORMAT()
- Pattern Matching with LIKE
- Type Conversion: CAST(), CONVERT(), TRY_CAST()
- Labs :
    - **Lab 3.1** – Write SELECT statements with aliases
    - **Lab 3.2** – Use WHERE, ORDER BY, DISTINCT, and TOP
    - **Lab 3.3** – Apply string, math, and date functions
    - **Lab 3.4** – Practice data type conversions and pattern matching with LIKE

Module 4: Intermediate Querying Techniques

- Multi-Table Joins: INNER, LEFT, RIGHT, FULL, and CROSS JOINs
- Join Conditions and Aliasing
- Aggregations: COUNT(), SUM(), AVG(), MIN(), MAX()
- GROUP BY and HAVING Clauses
- Subqueries: Scalar, Table-Based, Correlated
- Set Operations: UNION, INTERSECT, EXCEPT
- Derived Tables and Table Aliases

- CASE Expressions and Conditional Logic
- APPLY Operators: CROSS APPLY and OUTER APPLY
- Labs :
    - **Lab 4.1** – Write INNER, LEFT, RIGHT, and FULL JOINs
    - **Lab 4.2** – Perform aggregations using GROUP BY and HAVING
    - **Lab 4.3** – Use scalar and correlated subqueries
    - **Lab 4.4** – Use UNION and EXCEPT with real-world scenarios
    - **Lab 4.5** – Implement CASE expressions with business logic

Module 5: Advanced Querying and SQL Patterns

- Common Table Expressions (CTEs) and Recursive Queries
- Window Functions:
    - ROW_NUMBER(), RANK(), DENSE_RANK()
    - LEAD(), LAG(), NTILE(), FIRST_VALUE(), LAST_VALUE()
    - PARTITION BY and ORDER BY in OVER()
- Pivoting and Unpivoting Data
- Merging Data with the MERGE Statement (Upserts)
- JSON Handling in SQL Server:
    - OPENJSON(), JSON_VALUE(), JSON_QUERY()
    - FOR JSON AUTO/PATH
- Labs :
    - **Lab 5.1** – Create and use Common Table Expressions (CTEs)
    - **Lab 5.2** – Write recursive CTEs
    - **Lab 5.3** – Use ROW_NUMBER, RANK, DENSE_RANK, and NTILE
    - **Lab 5.4** – Apply PIVOT/UNPIVOT for data reshaping
    - **Lab 5.5** – Use JSON_VALUE and OPENJSON
    - **Lab 5.6** – Practice MERGE to handle insert/update logic

Module 6: Views, Indexes, and Triggers

- Creating and Managing Views
    - Simple Views and WITH CHECK OPTION
    - Updating Data through Views
- Indexing for Query Performance

    - Clustered vs Non-Clustered Indexes
    - Filtered and Covering Indexes
    - Execution Plans and Index Recommendations
- Triggers for Automation and Auditing
    - AFTER INSERT, UPDATE, DELETE
    - INSTEAD OF Triggers and Nested Triggers
- Labs :
    - **Lab 6.1** – Create simple and complex Views
    - **Lab 6.2** – Build Indexed Views
    - **Lab 6.3** – Create clustered and non-clustered indexes
    - **Lab 6.4** – Write AFTER and INSTEAD OF Triggers for audit use cases
    - **Lab 6.5** – Explore how triggers can interfere with business logic

Module 7: Query Optimization and Performance Tuning

- Reading and Interpreting Execution Plans
- Identifying Bottlenecks: Table Scan vs Index Seek
- Refactoring Queries for Better Performance
- Using SET STATISTICS TIME/IO
- Query Hints and Best Practices
- Labs :

Lab 7.1 – Compare performance of queries with/without indexes
Lab 7.2 – Analyze execution plans
Lab 7.3 – Use SET STATISTICS IO and TIME
Lab 7.4 – Refactor poorly written queries to better versions
Lab 7.5 – Try hints and test indexing strategies on large datasets

Module 8: Practical Use Cases

- Developing Sample Databases (HR, E-commerce, Sales)
- Constructing Complex Queries from Business Requirements
- Hands-on Scenarios: Reporting, Analysis, Data Cleanup
- Troubleshooting SQL Errors and Fixing Logic Flaws
- Labs :

- **Lab 8.1** – Simulate reporting queries with joins and filters
- **Lab 8.2** – Solve business case with aggregation and conditional logic
- **Lab 8.3** – Develop a reusable view for reporting
- **Lab 8.4** – Tune a poorly performing query with indexing and logic changes
- **Lab 8.5** – Handle errors and NULLs gracefully in complex queries