Writing SQL Queries

The course describes the ANSI/ISO SQL standard, but also identifies deviations from the standard in the two most widely used database products, Oracle and Microsoft SQL Server.

Course Agenda

Please note that this list of topics is based on our standard course offering, evolved from typical industry uses and trends. We’ll work with you to tune this course and level of coverage to target the skills you need most. Topics, agenda and labs are subject to change, and may adjust during live delivery based on audience needs, participation and skill-level.

1. **SQL Quick Refresher**
* SQL fundamentals
* Why SQL can be both easy and difficult
* Recommendations for thorough testing
1. **Retrieving data with SELECT**
* Expressions
* Literals
* Handling NULLs properly
1. **Executing queries**
* Analyzing query plans
* Enhancing query performance
* Retrieving partly results with FETCH and OFFSET
* Selecting the best alternatives
* Avoiding errors and pitfalls
* Querying Multiple Tables
1. **Implementing various types of joins**
* Inner joins
* Cross joins
* Left, right and full outer joins
* Equijoins vs. theta joins
* The performance implications of joins
* Adding filter conditions to outer joins
1. **Writing self joins**
* Joining a table to itself
* Chaining self joins
* Solving time-interval problems
1. **Combining queries with set operators**
* UNION
* UNION ALL
* INTERSECT
* EXCEPT
* Aggregate Functions
1. **Summarizing data with aggregate functions**
* COUNT
* SUM
* AVG
* MIN
* MAX
* Managing NULLs
* identifying duplicates
1. **Grouping data**
* GROUP B
* Applying conditions with HAVING
* Calculating moving averages
* Building crosstab reports
1. **Extending group queries**
* Nesting grouped aggregates
* Joins and grouping
* Introducing subtotals with CUBE and ROLLUP
* Performing Extensive Analysis with Analytic Functions
1. **The OVER clause**
* Specifying the ordering before applying the function
* Splitting the result set into logical partitions
* Calculating ranks
* RANK and DENSE\_RANK
* ROW\_NUMBER with ordered sets
* Calculating percentiles
1. **Extending the use of aggregates**
* Partitioning in multiple levels
* Computing running totals
* Comparing row and aggregate values
* Top-N queries
* Defining sliding window boundaries
1. **Building Subqueries**
2. **Self-contained subqueries**
* Subqueries in conditions and column expressions
* Creating multilevel subqueries
* Avoiding problems when subqueries return NULLs
* Handling multirow subquery results
* Finding gaps in number series
1. **Correlated subqueries**
* Accessing values from the outer query
* EXISTS vs. IN
* Identifying duplicates
* Avoiding accidental correlation
1. **Common table expressions**
* Reusable subqueries
* Recursive subqueries
* Traversing hierarchies
* Breaking Down Complex Queries
* Overcoming SQL limitations
* Reducing complexity and improving performance