Faysal Aziz

SQL for Practitioners

From Basic to Advanced in One Day (or 2x half days)

FA-0423 · One Day

Data & Analytics

Introduction

In today's data-rich world, the ability to query, manipulate, and secure relational data is not optional—it's foundational. Whether you're building applications, supporting data workflows, or performing analytics, fluency in SQL and database execution methods enables you to work efficiently, reliably and securely. In this one-day intensive course, you'll go beyond academic treatments of SQL and drill into real-industry patterns: writing and executing queries, embedding SQL in code, dealing with database-specific syntax, implementing DDL, enforcing security and avoiding concurrency pitfalls like deadlocks. Your instructor brings more than three decades of hands-on experience in enterprise database systems and will use use-cases, best practices and live code rather than just theory.

Learning outcomes

By the end of this course you will be able to:

- Write and execute basic SQL statements (SELECT, INSERT, UPDATE, DELETE) with confidence.
- Understand how SQL is executed programmatically (for example via JDBC or other database drivers) and use tools (e.g., SQL*Plus or similar) for issuing SQL directly.
- Construct and optimise more complex SQL: inner joins, outer joins, subqueries, set operations, and database-dependent syntax.
- Create and execute DDL statements (CREATE TABLE, ALTER TABLE, DROP, indexes, constraints) and understand schema design implications.
- Understand and implement database security best-practices: user/role privileges, access control, encryption basics.
- Identify, understand and avoid concurrency and locking issues — including deadlocks — and apply practical mitigation strategies.

- Adapt your SQL and execution methods to different database dialects and environments (i.e., vendor-dependent syntax and features).

Prerequisites

- Basic familiarity with relational database concepts (tables, rows, columns, keys).
- Some programming or scripting experience (so you can follow examples of JDBC or similar).
- Access to a relational database environment (or ability to install/use one during the session).
- A willingness to engage with code and live SQL rather than purely conceptual slides.

Training outline

1. Introduction & Setup

- Brief overview of relational databases in 2025: why SQL still matters, trends (e.g., cloud, AI integration)
- Environment setup: accessing a database, connecting via command-line tools (e.g., SQL*Plus, psql, equivalent)
- Setting up JDBC (or another language driver) to issue SQL programmatically: connection, statement execution, result-handling

2. Fundamental SQL Syntax

- SELECT: simple queries, column selection, filtering (WHERE), sorting (ORDER BY), limiting results
- INSERT, UPDATE, DELETE: modifying data, transactional implications (BEGIN, COMMIT, ROLLBACK)
- Basic functions and expressions (aggregation: COUNT, SUM, AVG; grouping: GROUP BY; HAVING)
- Using tools: executing SQL via command-line tool vs programmatic driver

3. Advanced Querying Techniques

- JOINs: inner join, left (outer) join, right outer join, full outer join; when each is appropriate
- Subqueries: scalar, correlated, in-list, EXISTS/NOT EXISTS
- Set operations: UNION, INTERSECT, EXCEPT (where supported)
- Window functions (if time permits) and analytical queries
- Vendor-specific syntax and extensions: understanding how dialects differ and why that matters (e.g., Oracle vs SQL Server vs PostgreSQL)
- Performance considerations: indexes, execution plans, avoiding common pitfalls

4. Schema Definition and Data Definition Language (DDL)

- CREATE TABLE: defining tables, data types, primary keys, foreign keys, constraints
- ALTER TABLE: adding/dropping columns, modifying constraints
- DROP TABLE / TRUNCATE TABLE: implications for data and performance
- Index creation, unique constraints, and when to use them
- Vendor-specific features for schema and optimization

5. Security and Permissions

- User/role management: creating users, granting and revoking privileges
- Object-level vs schema-level vs database-level permissions
- Principle of least privilege in real environments
- Encryption, data-masking and auditing basics (especially relevant in current database trend contexts)
- Best practices for securing programmatic connections (via JDBC or equivalent) and connection strings

6. Concurrency, Locking and Deadlock Avoidance

- Understanding transactions: isolation levels, locks, latches
- How different isolation levels affect concurrency and data integrity
- Deadlocks: what they are, how they occur, how to detect (logs, trace), how to avoid (lock order, smaller transactions, appropriate indexing)
- Vendor-specific concurrency features and mitigation tactics (e.g., SQL Server improved locking in recent versions)

7. Programmatic Execution Methods – Deep Dive

- Using JDBC (or another relevant API) to issue SQL: connection setup, executing queries, executing updates, handling exceptions
- Using command-line tools (e.g., SQL*Plus, psql) vs GUI tools: pros/cons in production and debugging scenarios
- Integration patterns: issuing SQL from application code, batching, prepared statements, parameterization to avoid SQL injection
- Error handling, transaction management from application side, concurrency concerns when multiple clients issue SQL

8. Putting It All Together – Hands-On Scenario

- Live lab: schema design, create tables, grant privileges, insert data
- Write a set of queries: simple SELECT/INSERT/UPDATE/DELETE, then more complex joins and subqueries
- Demonstrate programmatic execution: via JDBC (or your chosen language) and via command line
- Simulate concurrency: two sessions/threads updating the same tables, observe locks, perhaps simulate a deadlock and then apply a fix
- Review vendor-specific syntax: pick a second dialect (e.g., Oracle vs SQL Server) and show differences, for example in outer join syntax or locking hints

---

This one-day course is intense and immersive—designed for professionals who need to hit the ground running with SQL in real application scenarios. The instructor's 30+ years of industry experience ensures that you will not just learn syntax, but see how SQL is used, mis-used and optimised in real systems.

© 2026 Faysal Aziz · Kuala Lumpur