

QA

INTERMEDIATE

5 DAYS

SQL Fundamentals

Build a strong foundation in database fundamentals and SQL by understanding core concepts, syntax, and data modeling principles. Learn how to design tables, query and join data, apply aggregations and subqueries, and address security and performance considerations for reliable database management.

Why SQL Fundamentals?

Reliable data validation

Query and verify data used by applications and tests.

Safer changes

Apply DML and constraints with integrity in mind.

Faster analysis

Filter, join, and aggregate data for QA and reporting needs.

Performance awareness

Use indexes and query patterns that scale.

What you'll be able to do

- ✓ Gain an insightful understanding of database fundamentals
- ✓ Learn the basics of SQL syntax
- ✓ Understanding data types and functions in SQL
- ✓ Learn how to Create and modify tables
- ✓ Learn the skill of Querying data from a database
- ✓ Understand how to join multiple tables in a database
- ✓ Learn how to Aggregate and summarize data
- ✓ Understand how to use subqueries and derived tables to retrieve data from a database
- ✓ Understanding data normalization for practical data analysis and validation tasks
- ✓ Understanding security and performance considerations

Who this is for

- Database Administrators
- Database Developers
- Business Analysts
- Data Scientists
- Data Engineers
- Data Analysts
- IT Professionals

Curriculum

01 SQL and Relational Foundations

- Apply sql and relational foundations using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

02 Querying and Filtering

- Apply querying and filtering using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

03 Sorting and Aggregation

- Apply sorting and aggregation using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

04 Joins and Subqueries

- Apply joins and subqueries using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

05 DML

- Apply dml using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

06 Views and Procedures

- Apply views and procedures using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

07 Indexes and Performance

- Apply indexes and performance using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

08 Advanced SQL

- Apply advanced sql using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

09 Analysis and Reporting

- Apply analysis and reporting using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

10 Security

- Apply security using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks

11 Best Practices

- Apply best practices using SQL in practical test workflows
- Build and run hands-on exercises with realistic scenarios
- Validate results, logs, and failure signals
- Capture reusable patterns for team frameworks