SQL Alchemy

Duration : 5 Days(40 hrs)

Pre-requisites : *Good Knowledge of SQL and working knowledge of python.*

Table Of Contents :

Day 1: Introduction to ORM and SQLAlchemy Basics

Module 1: Introduction to ORM Concepts

- Understanding Object-Relational Mapping (ORM) and its benefits
- Overview of SQLAlchemy and its role in Python- SQL integration
- Setting up SQLAlchemy in a Python environment

Module 2: SQLAlchemy Core Fundamentals

- Introduction to SQLAlchemy Core
- Creating database connections and engine configuration
- Executing basic SQL queries using SQLAlchemy Core

Day 2: Querying and Advanced SQLAlchemy Core

Module 3: Advanced SQLAlchemy Core

- Constructing complex SQL statements with SQLAlchemy Core

- Using expressions, functions, and SQL constructs
- Handling transactions and connection management in SQLAlchemy Core

Module 4: SQLAlchemy ORM Introduction

- Overview of SQLAlchemy ORM and its relationship to SQLAlchemy Core
- Declaring models and mapping database tables to Python classes
- CRUD operations with SQLAlchemy ORM: creating, reading, updating, and deleting data

Day 3: Relationships and Advanced ORM Techniques

Module 5: Working with Relationships

- Understanding relationships in SQLAlchemy ORM: one-to-one, one-to-many, many-to-many
- Defining and navigating relationships between ORM entities
- Cascading operations and handling relationship persistence

Module 6: Advanced ORM Techniques

- Querying related objects using joins and eager loading
- Customizing queries with advanced filtering and sorting techniques
- Performance optimization strategies for ORM queries

Day 4: Transactions, Concurrency, and ORM Best Practices

Module 7: Transactions and Concurrency Control

- Working with transactions in SQLAlchemy ORM
- Managing concurrent access and isolation levels
- Using versioning and optimistic locking in SQLAlchemy

Module 8: Best Practices and Optimization

- Best practices for designing ORM models and relationships
- Optimizing database schema design and indexing
- Testing and debugging SQLAlchemy-based applications

Day 5: Integration, Performance Tuning, and Project Work

Module 9: Integrating SQLAlchemy with Web Frameworks

- Integrating SQLAlchemy with Flask or Django frameworks
- Using SQLAlchemy in web applications: routing requests and handling responses
- Implementing RESTful APIs with SQLAlchemy ORM

Module 10: Performance Tuning and Beyond

- Caching strategies and query optimization techniques
- Monitoring and profiling SQLAlchemy applications
- Advanced topics: custom types, reflection, and database migrations with SQLAlchemy