Python with Multithreading, Parallel, Memory Management and Advanced DB Optimisation

Duration: 3 days / 24 hours
Prerequisites: Working Knowledge of Python Programming

Day 1: Multithreading and Parallel Programming in Python

Module 1: Concurrency and Parallelism in Python

- Understanding concurrency vs parallelism
- Python's Global Interpreter Lock (GIL)
- Identifying scenarios for multithreading vs multiprocessing

Module 2: Threading and Synchronization

- Creating and managing threads using threading module
- Thread synchronization tools: Locks, Semaphores, Conditions
- Producer-consumer pattern using thread-safe queues

Module 3: Multiprocessing Techniques

- Using multiprocessing for parallel execution
- Sharing data between processes
- Real-world examples like parallel file processing

Module 4: High-Level Concurrency with concurrent.futures and asyncio

- Using ThreadPoolExecutor and ProcessPoolExecutor
- Introduction to async programming with asyncio
- Comparing use cases for threading, multiprocessing, and asyncio

---

Day 2: Memory Management in Python

Module 5: Python's Memory Architecture

- Stack vs Heap memory
- Object memory allocation and reference counting
- Python object life cycle

Module 6: Garbage Collection and Leak Management

- Internal workings of Python's garbage collector

- Managing and debugging circular references
- Using the gc module for fine control

Module 7: Memory Profiling and Analysis

- Tools: sys, tracemalloc, objgraph, memory_profiler
- Techniques to detect and fix memory issues
- Case study: Profiling a sample Python script

Module 8: Writing Memory-Efficient Code

- Choosing the right data structures
- Using generators, __slots__, and memory-efficient libraries
- Practical code optimization for reduced memory usage

Day 3: Advanced DB Query Optimization and Performance Tuning with Python

Module 9: Writing High-Performance Queries

- Best practices for efficient SQL query writing
- Reading and interpreting query execution plans
- Using raw SQL vs ORM with performance in mind

Module 10: Indexing and Query Optimization

- Understanding primary and secondary indexes
- B-Tree and Hash indexing concepts
- Applying indexes for performance gains

Module 11: Python DB Interaction Optimization

- Efficient use of Python DB libraries (psycopg2, sqlalchemy)
- Connection pooling and async DB operations
- Transaction handling and batch processing

Module 12: End-to-End Performance Tuning Workshop

- Analyzing and profiling a sample Python-DB app
- Applying learned techniques: indexing, queries, profiling
- Hands-on optimization and tuning challenge