Java, Spring Boot & PostgreSQL Interview Roadmap

Target: 3–5 Years Experience | Product & Service Companies

1. Core Java (High Priority)

- OOP Principles: Encapsulation, Inheritance, Polymorphism, Abstraction
- Abstract Class vs Interface
- Method Overloading vs Overriding
- final and static keywords
- Immutability and String internals
- equals() and hashCode() contract
- Shallow copy vs Deep copy
- Collections Framework overview
- ArrayList vs LinkedList
- HashMap internal working (hashing, collision, resizing)
- ConcurrentHashMap vs HashMap
- Java 8 features: Lambda, Stream API, Functional Interfaces, Optional
- Exception Handling: checked vs unchecked, custom exceptions
- Multithreading basics: Thread lifecycle, synchronized, volatile
- ExecutorService, Callable vs Runnable

2. Spring Boot (Production Knowledge)

- Spring IOC Container and Dependency Injection
- Bean lifecycle and scopes
- Annotations: @Component, @Service, @Repository
- Constructor Injection vs Field Injection
- Spring Boot auto-configuration
- application.properties / application.yml
- Profiles and environment configuration
- REST API design using @RestController
- HTTP methods and status codes
- DTO vs Entity separation
- Spring Data JPA basics
- Entity relationships (@OneToMany, @ManyToOne)
- Lazy vs Eager loading
- JPQL vs Native Queries
- Pagination and Sorting
- N+1 problem and solutions
- @Transactional and transaction propagation
- JWT based authentication flow

- Global exception handling using @ControllerAdvice
- Bean validation using annotations

3. PostgreSQL & SQL (Very Important)

- Basic SQL commands: SELECT, INSERT, UPDATE, DELETE
- WHERE clause, ORDER BY, LIMIT, OFFSET
- Joins: INNER JOIN, LEFT JOIN, usage scenarios
- Aggregation functions: COUNT, SUM, AVG
- GROUP BY and HAVING
- Indexes: purpose, benefits, drawbacks
- Primary key vs Unique index
- ACID properties
- Transactions: commit and rollback
- Isolation levels (basic understanding)
- Schema design and normalization
- One-to-many and many-to-many relationships

4. Suggested Study Strategy

- Daily: 1 hour Java or Spring Boot (alternate days)
- Weekend: PostgreSQL and SQL practice
- Revise notes weekly instead of watching random videos
- Practice explaining concepts aloud as in interviews
- Relate every topic to your real project experience