**Câu hỏi phỏng vấn Java và cơ sở dữ liệu (P2)**

1. **Giải thích khái niệm "convention over configuration" và tại sao nó được áp dụng trong Spring Boot.**
2. **Hãy giải thích sự khác biệt giữa @Component, @Service, @Repository và @Controller trong Spring Boot.**
3. **Làm thế nào để cấu hình cơ sở dữ liệu trong ứng dụng Spring Boot?**
4. **Giải thích cách Spring Boot quản lý các phụ thuộc (dependencies) và quy trình tự động cấu hình (auto-configuration).**
5. **Làm thế nào để xử lý các yêu cầu RESTful API trong Spring Boot?**
6. **Hãy nói về Spring Data JPA và cách bạn sử dụng nó trong ứng dụng Spring Boot.**
7. **Làm thế nào để xử lý xác thực và xác thực người dùng trong Spring Boot?**
8. **Giải thích khái niệm Spring Boot và tại sao nó được sử dụng trong phát triển ứng dụng Java backend.**
9. **Làm thế nào để tạo một ứng dụng Spring Boot và cấu hình môi trường phát triển (development environment)?**
10. **Giải thích khái niệm "convention over configuration" trong Spring Boot và cách nó giúp đơn giản hóa quá trình phát triển.**
11. **Hãy giải thích sự khác biệt giữa @Component, @Service, @Repository và @Controller trong Spring Boot và khi nào nên sử dụng chúng.**
12. **Làm thế nào để xử lý các yêu cầu RESTful API trong Spring Boot, bao gồm cách ánh xạ URL và phương thức HTTP với các phương pháp xử lý tương ứng.**
13. **Hãy giải thích cách Spring Boot quản lý các phụ thuộc (dependencies) thông qua Maven hoặc Gradle.**
14. **Làm thế nào để cấu hình và kết nối với cơ sở dữ liệu trong ứng dụng Spring Boot, ví dụ: MySQL, PostgreSQL.**
15. **Giải thích cách sử dụng Spring Data JPA trong Spring Boot để thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu.**
16. **Làm thế nào để xử lý xác thực và xác thực người dùng trong ứng dụng Spring Boot, bao gồm việc sử dụng Spring Security.**
17. **Giải thích khái niệm cơ sở dữ liệu quan hệ (relational database) và cách nó được sử dụng trong phát triển ứng dụng Java backend.**
18. **Hãy nêu và giải thích các thành phần chính của một câu lệnh SQL (Structured Query Language) và cách sử dụng chúng trong truy vấn cơ sở dữ liệu.**
19. **Làm thế nào để thiết kế cơ sở dữ liệu quan hệ hiệu quả, bao gồm việc xác định các bảng, quan hệ và các khóa chính (primary key) và khóa ngoại (foreign key)?**
20. **Giải thích khái niệm indexing và lợi ích của việc sử dụng index trong cơ sở dữ liệu.**
21. **Hãy giải thích sự khác biệt giữa INNER JOIN, LEFT JOIN và RIGHT JOIN trong câu lệnh SQL và cách sử dụng chúng.**
22. **Làm thế nào để tối ưu hóa hiệu suất truy vấn cơ sở dữ liệu, bao gồm sử dụng chỉ mục (index), tinh chỉnh câu lệnh SQL và tối ưu hóa schema.**
23. **Hãy nói về quy trình sao lưu và phục hồi dữ liệu trong cơ sở dữ liệu, bao gồm việc sử dụng transaction log và công cụ sao lưu (backup) như MySQL Dump.**
24. **Làm thế nào để xử lý và bảo mật thông tin nhạy cảm như mật khẩu trong cơ sở dữ liệu, bao gồm việc sử dụng hash và salt.**
25. **Hãy giải thích sự khác biệt giữa cơ sở dữ liệu quan hệ và cơ sở dữ liệu NoSQL và cách sử dụng chúng trong ứng dụng Java backend.**
26. **Làm thế nào để thực hiện các thao tác CRUD (Create, Read, Update, Delete) với cơ sở dữ liệu trong ứng dụng Java backend, bao gồm việc sử dụng JDBC hoặc ORM framework như Hibernate.**
27. **Làm thế nào để thực hiện kết nối và tương tác với cơ sở dữ liệu trong ứng dụng Java backend sử dụng JDBC?**
28. **Hãy giải thích khái niệm ORM (Object-Relational Mapping) và lợi ích của việc sử dụng ORM framework như Hibernate trong phát triển ứng dụng Java backend.**
29. **Làm thế nào để thực hiện các thao tác CRUD với cơ sở dữ liệu bằng cách sử dụng Hibernate?**
30. **Hãy nêu và giải thích các mối quan hệ giữa các thực thể (entities) trong mô hình dữ liệu và cách bạn ánh xạ chúng vào cơ sở dữ liệu trong Hibernate.**
31. **Làm thế nào để tối ưu hóa hiệu suất truy vấn trong Hibernate, bao gồm sử dụng caching, lazy loading và fetching strategy?**
32. **Hãy giải thích cách sử dụng transaction trong ứng dụng Java backend và tại sao nó quan trọng trong môi trường đa luồng và có nhiều người dùng.**
33. **Làm thế nào để xử lý và kiểm soát lỗi trong quá trình tương tác với cơ sở dữ liệu trong ứng dụng Java backend?**
34. **Hãy giải thích cách thực hiện việc migrate dữ liệu (database migration) trong môi trường phát triển và triển khai ứng dụng Java backend.**
35. **Làm thế nào để đảm bảo tính toàn vẹn dữ liệu trong cơ sở dữ liệu, bao gồm sử dụng ràng buộc (constraints) và các quy tắc xử lý dữ liệu (data validation rules)?**
36. **Hãy nói về quy trình sao lưu và khôi phục dữ liệu, bao gồm việc sử dụng transaction log và công cụ như MySQL Dump.**