**REST API Quiz: Entities, CRUD Operations, Annotations, Commands, and MySQL**

**Target Audience**: 2nd-Year Students  
**Total Questions**: 15

**Questions and Answers**

1. **Direct**:  
   What annotation is used in Spring Boot to expose a method as a REST API endpoint?  
   **a)** @Service  
   **b)** @RestController  
   **c)** @Repository  
   **d)** @RequestMapping  
   **Answer**: **b)** @RestController
2. **Direct**:  
   Which annotation is used to bind path variables in a URL to method parameters in Spring Boot?  
   **a)** @PathVariable  
   **b)** @RequestParam  
   **c)** @RequestBody  
   **d)** @Entity  
   **Answer**: **a)** @PathVariable
3. **Scenario-Based**:  
   To fetch a list of employees based on their department, which SQL query would you use?  
   **a)** SELECT \* FROM employees WHERE department = 'HR';  
   **b)** UPDATE employees SET department = 'HR';  
   **c)** DELETE FROM employees WHERE department = 'HR';  
   **d)** INSERT INTO employees (department) VALUES ('HR');  
   **Answer**: **a)** SELECT \* FROM employees WHERE department = 'HR';
4. **Direct**:  
   What annotation is used to define a primary key in a JPA entity?  
   **a)** @Entity  
   **b)** @Id  
   **c)** @GeneratedValue  
   **d)** @Table  
   **Answer**: **b)** @Id
5. **Scenario-Based**:  
   A client sends a POST request to /products with product details. How would you represent the inserted product in the database using SQL?  
   **a)** INSERT INTO products (name, price) VALUES ('Phone', 1000);  
   **b)** SELECT \* FROM products;  
   **c)** DELETE FROM products WHERE name = 'Phone';  
   **d)** UPDATE products SET price = 1000 WHERE name = 'Phone';  
   **Answer**: **a)** INSERT INTO products (name, price) VALUES ('Phone', 1000);
6. **Direct**:  
   What is the purpose of the @RequestBody annotation in Spring Boot?  
   **a)** To map HTTP request body to a Java object  
   **b)** To bind query parameters to method arguments  
   **c)** To define a database entity  
   **d)** To log HTTP requests  
   **Answer**: **a)** To map HTTP request body to a Java object
7. **Direct**:  
   Which command is used to create a database in MySQL?  
   **a)** CREATE TABLE database\_name;  
   **b)** CREATE DATABASE database\_name;  
   **c)** USE database\_name;  
   **d)** INSERT INTO database\_name;  
   **Answer**: **b)** CREATE DATABASE database\_name;
8. **Scenario-Based**:  
   You want to ensure that the email column in a users table is always unique. Which SQL command would you use?  
   **a)** ALTER TABLE users ADD CONSTRAINT email\_unique UNIQUE (email);  
   **b)** SELECT DISTINCT email FROM users;  
   **c)** CREATE TABLE users (email VARCHAR(255) UNIQUE);  
   **d)** Both a and c  
   **Answer**: **d)** Both a and c
9. **Direct**:  
   Which annotation is used to create a repository in Spring Boot?  
   **a)** @Controller  
   **b)** @RestController  
   **c)** @Repository  
   **d)** @Service  
   **Answer**: **c)** @Repository
10. **Scenario-Based**:  
    To update a user's phone number in the database, which SQL query would you use?  
    **a)** UPDATE users SET phone = '1234567890' WHERE id = 1;  
    **b)** SELECT \* FROM users WHERE phone = '1234567890';  
    **c)** INSERT INTO users (phone) VALUES ('1234567890');  
    **d)** DELETE FROM users WHERE phone = '1234567890';  
    **Answer**: **a)** UPDATE users SET phone = '1234567890' WHERE id = 1;
11. **Direct**:  
    What is the default port for a Spring Boot application?  
    **a)** 3306  
    **b)** 8080  
    **c)** 1521  
    **d)** 8000  
    **Answer**: **b)** 8080
12. **Scenario-Based**:  
    To delete all rows from a logs table in MySQL, but keep the table structure, which command would you use?  
    **a)** DROP TABLE logs;  
    **b)** DELETE FROM logs;  
    **c)** TRUNCATE TABLE logs;  
    **d)** ALTER TABLE logs;  
    **Answer**: **c)** TRUNCATE TABLE logs;
13. **Direct**:  
    Which annotation in Spring Boot specifies a class as a service layer component?  
    **a)** @Controller  
    **b)** @RestController  
    **c)** @Service  
    **d)** @Component  
    **Answer**: **c)** @Service
14. **Scenario-Based**:  
    A database table students contains the columns id, name, and age. How would you find all students older than 20?  
    **a)** SELECT \* FROM students WHERE age > 20;  
    **b)** SELECT age FROM students WHERE age > 20;  
    **c)** SELECT id, name FROM students WHERE age > 20;  
    **d)** All of the above  
    **Answer**: **d)** All of the above
15. **Direct**:  
    Which annotation is used to configure database relationships in JPA?  
    **a)** @ManyToOne  
    **b)** @OneToMany  
    **c)** @ManyToMany  
    **d)** All of the above  
    **Answer**: **d)** All of the above

Would you like this quiz formatted for a specific platform (e.g., printable PDF, LMS)?