Spring Boot Interview Questions & Answers

Chapter 15: Spring Boot – With Database

Q: How does Spring Boot connect to a database?

Spring Boot connects to databases using JDBC or JPA with the help of auto-configuration. You just need to provide database details in application.properties, and Spring Boot sets up the DataSource automatically.

Use-Case / Example:
spring.datasource.url=jdbc:mysql://localhost:3306/school
spring.datasource.username=root
spring.datasource.password=pass

Q: What is Spring Data JPA and why is it used?

Spring Data JPA is a module that simplifies database access by providing repository interfaces for CRUD operations. It reduces boilerplate code and makes database interactions easy.

Use-Case / Example: Instead of writing SQL queries, you can just create an interface StudentRepository extends JpaRepository and use built-in methods like findAll() or save().

Q: How do you define an entity in Spring Boot?

An entity is a class mapped to a database table using @Entity annotation. Each field is mapped to a column.

Use-Case / Example:
@Entity
public class Student {
@Id
private int id;
private String name;
}

Q: How do repositories work in Spring Boot?

Repositories are interfaces that extend Spring Data JPA's JpaRepository or CrudRepository. Spring Boot automatically provides implementations at runtime.

Use-Case / Example:
public interface StudentRepository extends JpaRepository {}
Now you can call studentRepository.findAll() without writing SQL.

Q: How do you configure H2 database in Spring Boot?

Spring Boot supports the in-memory H2 database for testing. You just add the dependency and configure it in application.properties.

Use-Case / Example:
spring.datasource.url=jdbc:h2:mem:testdb
spring.h2.console.enabled=true
This enables a web console at /h2-console.

Q: What is the difference between JPA and JDBC?

- JDBC → requires manual SQL queries.
- JPA → abstracts database operations using objects and repositories.

Use-Case / Example: With JDBC, you write SELECT * FROM students. With JPA, you just call studentRepository.findAll().

Q: How do you run custom SQL queries in Spring Boot?

You can use the @Query annotation in repositories to define custom SQL or JPQL queries.

Use-Case / Example:
@Query("SELECT s FROM Student s WHERE s.name = :name")
List findByName(@Param("name") String name);

Q: How do you handle transactions in Spring Boot?

You use @Transactional to ensure a set of database operations run as a single unit. If one operation fails, the transaction rolls back.

Use-Case / Example: In a banking app, transferring money involves debiting one account and crediting another. If one fails, both should roll back.

Q: How can you switch databases between environments?

You can use different profiles (application-dev.properties, application-prod.properties) with different database configurations.

Use-Case / Example:
- Dev → H2 database for quick testing
- Prod → MySQL or PostgreSQL for real data

Q: Why is Spring Boot widely used with databases?

Because it provides auto-configuration, easy integration with JPA, support for multiple databases, and reduces boilerplate code. It makes database development much faster and easier.

Use-Case / Example: A startup building an e-commerce app can quickly set up MySQL with Spring Boot and use JPA repositories to handle orders, products, and customers.