Beaconfire Inc, Home Work, Week3 Day13.
By (Ping) Nalongsone Danddank.
ndanddank@gmail.com
wechatID: ndanddank
Short Answer:

1. What is Spring Boot? How does it differ from Spring?

-> Spring Boot makes it easy to create stand-alone, production-grade Spring-based Applications that you can just run. Benefits:

• It's easy to develop Spring based application
• Spring Boot takes less time, improve overall productivity
• No need to write templet code, XML configuration or redundant annotation
• Easy to integrate with other Spring feature such as Spring JDBC, Spring ORM, Spring AOP, Spring Security and etc.
• Provides embedded HTTP server such as Tomcat, easy for web app development and test.

Spring	Spring Boot
Widely used for building enterprise Java applications	Widely used for building REST APIs
Aims to simplify enterprise Java development	Aims to shorten code length and easily build web applications
Allows building loosely coupled applications	Allows building standalone applications
Main feature is dependency injection	Main feature is auto-configuration
Involves writing lots of boilerplate code	Reduces boilerplate code
Needs dependencies to be defined manually	Starters take care of dependencies
Involves setting up server manually	Includes embedded server like Tomcat and Jetty

2. What are JDBC statements? List all types of JDBC statements and their usage.

-> JDBC stands for "Java Database Connectivity". It is an API (Application Programming Interface) which consists of a set of Java classes, interfaces and exceptions and a specification to which both JDBC driver vendors and JDBC developers (like us) adhere when developing applications.

-> Three interfaces are used for sending SQL statements to databases:

1.Statement:

• A Statement object is used to send a simple SQL statement to the database with no parameters
• If n rows need to be inserted, then the same statement gets compiled n number of times

2.PreparedStatement:

• A PreparedStatement object sends precompiled statements to the databases with or without IN parameters
• Only the values that have to be inserted are sent to the database again and again
• Increase efficiency

3.CallableStatement:

• A CallableStatement object is used to call stored procedures
• Better performance due to pre-compilation

3. What is Database Transaction?
-> Transaction represents a single unit of work.

4. What is ACID?
• The ACID properties describes the transaction management well.
• Atomicity: means either all successful or none.
• Consistency: ensures bringing the database from one consistent state to another consistent state.
• Isolation: ensures that transaction is isolated from other transaction.
• Durability: means once a transaction has been committed, it will remain, even in the event of errors, power loss etc.

5. How do we usually perform Transaction Management in JDBC?
-> JDBC allows SQL statements to be grouped together into a single transaction
• Transaction control is performed by the Connection object,
• default mode is auto-commit, I.e., each sql statement is treated as a transaction
• We can turn off the auto-commit mode with con.setAutoCommit(false);
• And turn it back on with con.setAutoCommit(true);
• Once auto-commit is off, no SQL statement will be committed until an explicit is invoked con.commit();
• At this point all changes done by the SQL statements will be made permanent in the database.

- In JDBC, **Connection interface** provides methods to manage transaction

Method	Description
void setAutoCommit(boolean status)	It is true by default means each transaction is committed by default.
void commit()	commits the transaction.
void rollback()	cancels the transaction.

#1) setAutoCommit() Method
By default, the value of AutoCommit value is TRUE. After the execution of the SQL statement, it will be committed automatically. By using the setAutoCommit() method we can set the value to AutoCommit.
#2) Commit() Method
The commit method is used to commit the data. After the execution of the SQL statement, we can call the commit(). It will commit the changes which are made by the SQL statement.
Syntax: conn.commit();

#3) Rollback() Method

The rollback method is used to undo the changes till the last commit has happened. If we face any issue or exception in the execution flow of the SQL statements, we may roll back the transaction.

Syntax: conn.rollback();

#4) setSavepoint() Method

Savepoint gives you additional control over the transaction. When you set a savepoint in the transaction (a group of SQL statements), you can use the rollback() method to undo all the changes till the savepoint or after the savepoint(). setSavepoint() method is used to create a new savepoint.

#5) releaseSavepoint() Method

It is used to delete the created savepoint.

When complete the transaction we have to close it.

https://www.softwaretestinghelp.com/jdbc-transaction-management/

6. What is JdbcTemplate? And what are some of the advantages it has over standard JDBC?

-> Spring JdbcTemplate is a powerful mechanism to connect to the database and execute SQL queries. It internally uses JDBC api, but eliminates a lot of problems of JDBC API.

Action	Spring	You
Define connection parameters.		X
Open the connection.	X	
Specify the SQL statement.		X
Declare parameters and provide parameter values		X
Prepare and execute the statement.	X	
Set up the loop to iterate through the results (if any).	X	
Do the work for each iteration.		X
Process any exception.	X	
Handle transactions.	X	
Close the connection, the statement, and the resultset.	X	