1)Thread Safety, how is this achieved?

2)Error vs Exception

3)Explain about DNS

4)HTTP Error Codes explain

5)What happens when you type URL in browser and press enter

6)What is virtual IP address

7)IPV6

8)SSL vs TLS

9)HTTP vs HTTPS

10)When multiple users login at sametime everday server response time is slow from 10am to 10.05am everday. What could be the reason?

11)WebHosting and WebTraffic

12)Authentication vs authorization

DDL vs DML commands

Insert,Delete,Update which is costly

Difference b/w functions and procedures

Statement vs PreparedStatement vs Callable Statement

What jar file used for jdbc connectivity

Types of drivers

How to delete one column in table

View vs Index

HTML vs javascript

How validation done and describe flow in application

Explain struts application flow

What are parameters in action servlet

What is JMS used for

Spring dependency injection,aop,ioc concepts

How did you configure log4j

What is JUNIT used for, what tool did you use

Difference b/w webserver and application server

When user logs into application describe the flow

Why Spring JDBC ?

Why Spring JDBC ?

The problems of JDBC API are as follows:

* We need to write a lot of code before and after executing the query, such as creating connection, statement, closing resultset, connection etc.
* We need to perform exception handling code on the database logic.
* We need to handle transaction.
* Repetition of all these codes from one to another database logic is a time consuming task.
* Spring JdbcTemplate eliminates all the above mentioned problems of JDBC API. It provides you methods to write the queries directly, so it saves a lot of work and time.
* Spring JDBC provides only abstract layer to eliminate the limitations of traditional JDBC. So application would still be using traditional JDBC API internally through the abstract layer of Spring JDBC.

Spring JDBC – Repository Class

@Repository annotation at class level.

Enables spring f/w autoscanning support to create repository bean without the explicit bean defining in the configuration.

This also causes Spring Exception translation(translates checked exceptions to unchecked exceptions)

Spring JDBC – JdbcTemplate Class

It takes care of creation and release of resources such as creating and closing of connection object etc. It handles the exception and provides the informative exception messages by the help of excepion classes defined in the **org.springframework.dao** package.We can perform all the database operations by the help of JdbcTemplate class such as insertion, updation, deletion and retrieval of the data from the database

**applicationContext.xml**

The **DriverManagerDataSource** is used to contain the information about the database such as driver class name, connnection URL, username and password.

<https://www.javatpoint.com/spring-JdbcTemplate-tutorial>

Spring NamedParameterJdbcTemplate

<https://www.javatpoint.com/spring-NamedParameterJdbcTemplate-example>

# Hibernate and Spring Integration

We can simply integrate **hibernate application with spring application**.

In hibernate framework, we provide all the database information hibernate.cfg.xml file.

But if we are going to integrate the hibernate application with spring, we don't need to create the hibernate.cfg.xml file. We can provide all the information in the applicationContext.xml file.

# Spring Data JPA Tutorial

Spring Data JPA API provides JpaTemplate class to integrate spring application with JPA.

JPA (Java Persistent API) is the sun specification for persisting objects in the enterprise application. It is currently used as the replacement for complex entity beans.

The implementation of JPA specification are provided by many vendors such as:

* Hibernate
* Toplink
* iBatis
* OpenJPA etc.

## Advantage of Spring JpaTemplate

You don't need to write the before and after code for persisting, updating, deleting or searching object such as creating Persistence instance, creating EntityManagerFactory instance, creating EntityTransaction instance, creating EntityManager instance, commiting EntityTransaction instance and closing EntityManager.