Day-14 Interview Questions

1.How do you execute an INSERT statement using the JDBC Statement interface?

To execute an INSERT statement using the Statement interface, you can create a Statement object, construct the SQL INSERT query as a string, and then use the executeUpdate() method to execute the query.

2.Explain the process of updating records in a database table using the Statement interface?

To update records using the Statement interface, you can create a Statement object, construct the SQL UPDATE query as a string, and execute it using the executeUpdate() method. Ensure that the query specifies the columns to update and sets new values.

3. How do you delete records from a database table using the Statement interface?

To delete records using the Statement interface, create a Statement object, construct the SQL DELETE query as a string, and execute it using the executeUpdate() method. Make sure the query specifies the criteria for record deletion (e.g., using a WHERE clause).

4. How can you handle SQLExceptions when executing SQL statements using the JDBC Statement interface?

SQLExceptions can occur when executing SQL statements with the Statement interface. You should use try-catch blocks to catch and handle these exceptions. For example:

5.What is the role of the executeUpdate() method in error handling when executing SQL statements using the Statement interface?

The executeUpdate() method not only executes SQL statements but also returns the number of rows affected by the statement. This information can be used for error handling. If executeUpdate() returns 0, it indicates that no rows were affected, which can be a useful error condition to check for updates or deletes.

6.Explain how you can ensure proper resource management (closing statements and connections) when handling errors in JDBC using the Statement interface.

To ensure proper resource management when handling errors with the Statement interface, use a try-with-resources block or explicitly close the statement and connection in a finally block. This ensures that resources are properly released, even if an exception occurs.

7.Explain the difference between checked and unchecked exceptions in JDBC error handling.

Checked exceptions (e.g., SQLException) must be explicitly caught or declared, while unchecked exceptions (e.g., NullPointerException) don't require explicit handling. JDBC primarily throws checked exceptions, which means you need to handle them using try-catch blocks or propagate them up the call stack.

8.What is connection pooling in JDBC, and how does it contribute to efficient error handling?

Connection pooling is a technique where database connections are maintained and reused, reducing the overhead of establishing new connections. It can contribute to efficient error handling by ensuring that connections are in a consistent state when errors occur, and they can be released and reused for subsequent operations.