Discussion Question

1.How exceptions are handled during the process:

In the provided Java code, exception handling is used to catch potential errors that might occur during database operations using JDBC. SQLExceptions are handled when creating the Employee table, inserting records, searching, updating, or deleting records. NumberFormatException is handled for parsing numeric input fields. The catch blocks display appropriate error messages to the user, ensuring they are aware of any issues that occurred during the database operations.

2.Main issues occurred during the JDBC and Swing connection process:

Some common issues that may occur during JDBC and Swing connection include:

Incorrect database URL, username, or password

Missing JDBC driver for the database

Incorrect SQL queries

Network or connectivity issues with the database server

Improperly handling database connections, leading to resource leaks

To avoid such issues, it's essential to properly handle exceptions, close connections after use, and thoroughly test the application with different scenarios.

3.How you can use threads in this program:

To enhance the responsiveness of the UI and avoid freezing the application during database operations, you can use threads. Specifically, you can use the SwingWorker class in Swing to perform time-consuming tasks in the background while keeping the UI responsive.

For example, when inserting, updating, or deleting records, you can create a SwingWorker instance to execute the database operation in the doInBackground() method. Meanwhile, you can show a progress bar or some visual indication to the user that the task is being processed. Once the task is completed, you can update the UI with the results.