Derrick Fox

CS 214 - Advanced Java

April 27, 2015

Project 10 - Batch Processing

Algorithm, UML Diagram and Screen Shots

Environment:

* This program assumes that you have a database already installed on your root directory named “javabook” with a username of “scott” and a password of “tiger” like in our textbook.
* This program includes the files “Main.java”, “DBConnectionPane.java” and “ComparePane.java”.
* This program assumes that the proper MySQL JAR files has been included and that the build-path has that JAR file included.
* This program creates a database table called “Temp” and if the table already exists, the program will drop that table and create a new Staff table.

**Main**:

start:

1. Create a new DBConnectionPane object.

**DBConnectionPane**:

1. Declare a Connection object to the database and all of the GUI elements.

connectDB:

1. Connect to the database using the declared Connection object populated by the GUI’s user inputs.
2. Switch the stage of the Pane by creating a new ComparePane object.

**ComparePane:**

1. Declare a Statement object
2. Create a SQL statement that creates a table or drops it if it exits.
3. Create GUI
4. Create event listeners for “Connect” button, “Batch Process” button, and “UnBatched Process” button.

unBatchProcess:

1. Declare three arrays to hold the three double values that will be sent to the database.
2. Create a long variable to hold the system’s nanoTime value start-time.
3. Create a for-loop to generate three random double values.
4. Create a for-loop to input values for an Insert statement and then send that statement for at the end of each loop.
5. Create a long variable to hold the system’s nanoTime value end-time minus the system’s nanoTime start-time.
6. Print the time difference to the stage’s text area.

batchProcess:

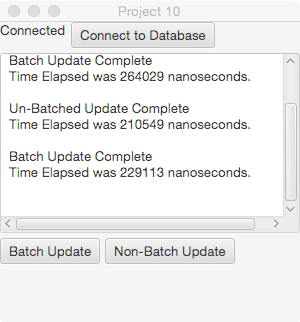
1. Declare three arrays to hold the three double values that will be sent to the database.
2. Create a long variable to hold the system’s nanoTime value start-time.
3. Create a for-loop to generate three random double values.
4. Create a for-loop to input values for an Insert statement and add that statement to the batch at the end of each loop.
5. After the loop, send the entire batch to the database.
6. Create a long variable to hold the system’s nanoTime value end-time minus the system’s nanoTime start-time.
7. Print the time difference to the stage’s text area.

**UMLs on next page…**

| Main |
| --- |
| - theStage: Stage |
| + start: void |
| + main: void |

| DBConnectionPane |
| --- |
| - connection: Connection |
| - lblConnectionStatus: Label |
| - btConnect: Button |
| - cboDriver: ComboBox |
| - cboURL: ComboBox |
| * tfUserName: TextField |
| - tfPassword: TextField |
| + DBConnectionPane: DBConnectionPane |
| + switchStage(): void |
| + connectDB(): void |

| ComparePane |
| --- |
| - statusLabel: Label |
| - connectionButton: Button |
| - reportTextArea: TextArea |
| - batchButton: Button |
| - unBatchButton: Button |
| - myStmt: Statement |
| + ComparePane(): ComparePane |
| + checkIfConnected(): void |
| + unBatchProcess(): void |
| + batchProcess(): void |

**Screen Shot**