CPRG 307
Assignment 2 (Modules 1 – 10)

Student: ______________________

Mark: _______ / 42

Assignment Instructions

The due date for this assignment is posted in D2L in the assignment submission area**.** Any assignment submitted after the due date will receive a mark of zero. To complete this assignment, follow the steps below:

☐ 1**.** WORK in a group of two (2) if you wish.

☐ 2**.** REVIEW the problem indicated in the *Problem* section below.

☐ 3. SUBMIT the tested coded solution in a zip file to the appropriate location in D2L.

Deliverables

☐ 1. SUBMIT the zip file containing your tested coded solution to the appropriate location in D2L. The zip file should have the following naming standard:

*lastname_firstname*_A2.zip

If working in a group of two (2), only one needs to submit to D2L (both can if you so wish). Both members will receive the same feedback. The zip file should have the following naming standard:

*lastname of member 1_lastname of member 2*_A2.zip

- Coded Solution:
  - Should include all PL/SQL scripts, SQL scripts, copy of SQL*Loader control file, and all Java source code.
    - Not including your Java source code in your submission will result in a zero (0) mark for all Java components identified in the *Evaluation* section.

☐ 2. DEMONSTRATE your GUI to your instructor by the day and time indicated by your instructor.

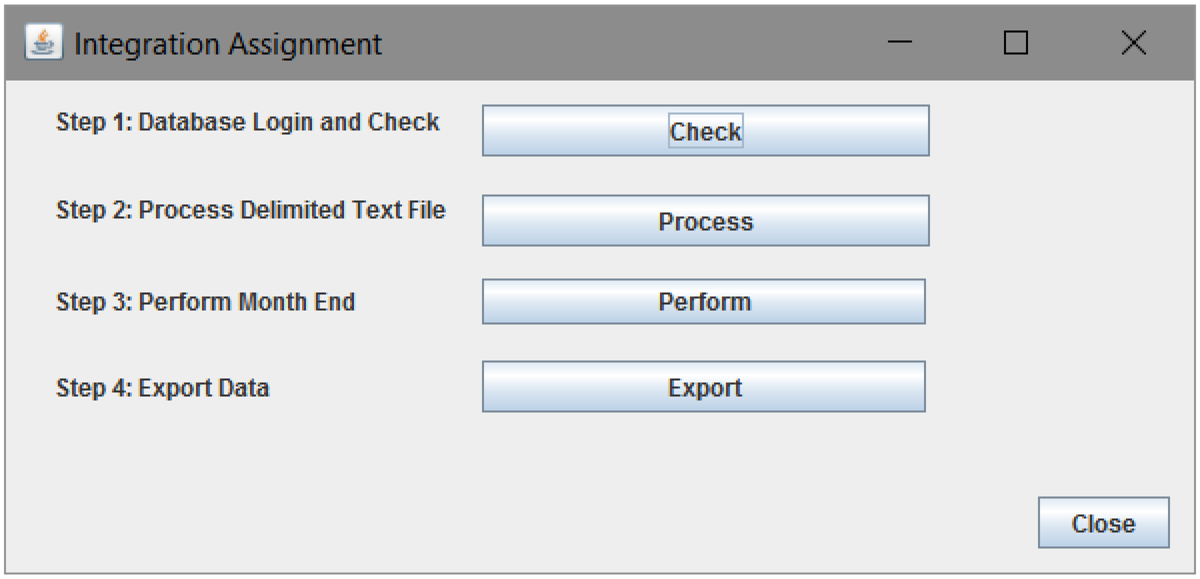
  - We are not looking for a functioning GUI at this time (this will be evaluated in your final submission)
  - This demo is just to see the GUI and is where the GUI marks will be given
    - If you do not demo, you will get zero (0) on the GUI as identified in the *Evaluation* section

Problem

This assignment is based on the We Keep It Storage (WKIS) Company's system, an accounting system, and is an extension from what we did in Assignment 1 (you do **not** have to have Assignment 1 completed and working to do Assignment 2). WKIS outsources its payroll to an external company called Payroll Systems Inc. (PSI). After each payroll run, PSI provides WKIS with a delimited text file with the payroll transaction information in it. WKIS needs this payroll information to maintain its accounting system transactions.

Your next task is to process this delimited text file. There are four distinctive steps (with multiple tasks each) that are performed and you will create a Java program (a GUI interface is required) that will handle these four steps. Your Java program will call SQL*Loader (an outside database utility), and call database stored procedures and functions. You will also create database triggers for some of the functionality.

An example GUI (yours can be different):


Integration Assignment
Step 1: Database Login and Check
Check
Step 2: Process Delimited Text File
Process
Step 3: Perform Month End
Perform
Step 4: Export Data
Export
Close

The four steps (guidelines for your application):

1) Connect to the database and check permissions.

    a. Prompt the user for database login credentials. The *username* and *password* should not be hard coded in your code.

    b. Connect to the database. If the user log in failed because of an incorrect username or password, simply display the error message to the screen to inform the user who can then try again if they so wish. No further action, other than your displaying of the error to the screen, is required.

    c. Create a stored function that checks if the user who has been logged into the database has permission to complete all of the steps.

        i. A user, when running this application, will require special permission before they can complete the last (fourth) step in this application. This permission is discussed later in this assignment document, but this permission is what we will be checking in order to see if the user should be able to continue.

        ii. This stored function will check that this user (the one currently connected to the database) has been given permission to *execute* the *UTL_FILE* package. When this function is called from your Java program, it will return *'Y'* if the user has this permission or it will return *'N'* if they do not.

            - To find out if the user has this permission, you will need to look at the USER_TAB_PRIVS data dictionary view. If the user has this permission, there will be an entry in this view for a privilege of *EXECUTE* for the package (in this case the column is TABLE_NAME) *UTL_FILE*.

                - 'EXECUTE' and 'UTL_FILE' can be hard coded <u>for this step</u>.

            - If *'N'* is returned by the function, your Java program should provide a message to the user indicating they do not have the required permissions to continue and the user should be disconnected from the database.

2) Process the delimited text file provided.

   a. A delimited text file has been given to you (available in D2L).

   b. This file (payroll.txt) has been created with a ";" as a delimiter.

   c. Using this delimited text file call the SQL*Loader utility from Java to load the data into the database. All new data should be placed in the PAYROLL_LOAD table.

      i. More information on the SQL*Loader utility can be viewed in the document provided in Module 9 on D2L.

   d. The SQL*Loader control file should be created dynamically to the location provided by the user when prompted. The control file should be created to use the name and location of the delimited file provided by the user.

   e. Each new payroll entry being loaded should be processed by the database. To do this a trigger (<u>one</u> DML trigger, not a compound trigger) should fire on the PAYROLL_LOAD table. This trigger will:

      i. Create a new transaction for each payroll entry and put this transaction into the NEW_TRANSACTIONS table. Remember that we are using double-entry accounting. For each row in PAYROLL_LOAD, a **complete transaction** must be added to the NEW_TRANSACTIONS table. This means NEW_TRANSACTIONS would need two entries – the first entry is the credit to the <u>accounts payable account</u> and the second entry is the debit to the <u>payroll expense account</u>.

         o As these two accounts will never change, you can hard code these account numbers <u>for this step</u>.

      ii. Every entry that goes into the NEW_TRANSACTIONS table will put a value into each column. Use the sequence available with the WKIS database to produce the transaction number.

      iii. The STATUS for the associated row in PAYROLL_LOAD should be changed to *G* if it was processed above with no errors or *B* if there were errors.

3) Perform the month end process.

   a. Your Java program will call a database stored procedure that will zero out all temporary accounts:

      i. Temporary accounts are considered the revenue and expense accounts.

         - These two account types ('RE', 'EX') can be hard coded <u>for this step</u>.

      ii. By "zero out", we do not mean changing the balance for these accounts to zero directly (i.e. the ACCOUNT table is never modified). We need to create a transaction (debit and credit) that will do this.

         - For any revenue account, to zero it out, create a transaction (placing it in the NEW_TRANSACTIONS table), with the revenue account having a debit for the balance amount and a credit to the <u>owners equity account</u> for the same amount.

            - The owner's equity account number can be hard coded <u>for this step</u>.

            - The revenue accounts should be dynamically retrieved from the database.

         - For any expense account, to zero it out, create a transaction (placing it in the NEW_TRANSACTIONS table), with the expense account having a credit for the balance amount and a debit to the <u>owners equity account</u> for the same amount.

            - The owner's equity account number can be hard coded <u>for this step</u>.

            - The expense accounts should be dynamically retrieved from the database.

      iii. Every entry that goes into the NEW_TRANSACTIONS table will put a value into each column. Use the sequence available with the WKIS database to produce the transaction number.

4) Export transactional data to a delimited file.

   a. Once all of the new transactions have been created, WKIS wants to be able to pull this data from the database and put it into a comma delimited file that will be used by an external application for analysis purposes.

   b. When the *Export* button is pressed on your GUI, your user should be prompted for three pieces of information:

      i. The complete directory path that the export file will go to.

      ii. The name of the delimited file to be created.

      iii. The "alias" for the directory path.

   c. Your Java program should then make a JDBC call using *executeUpdate*, like in the second semester Java course, to create the directory "alias". The command is as follows (CSV_DIR is the "alias" provided by the user).

      i. EG: `CREATE OR REPLACE DIRECTORY CSV_DIR AS '/home/oracle/Desktop/A2';`

         - Do not hard code the directory path. This should be the value given to you by the user when prompted.

d. Your Java program will then call a stored PL/SQL procedure to populate the delimited file.

   i. The store procedure should take in two parameters (the directory alias and the file name).

      - Programming Note: Inside the procedure, the directory alias (the value of the input parameter, not the parameter) must be in upper case only or it will not work correctly.

   ii. The delimiter that should be used is a comma.

   iii. The table to be exported is the NEW_TRANSACTIONS table.

   iv. All columns and rows for this table should be included in this export.

   v. To create and populate the file correctly, look at the UTL_FILE package: http://docs.oracle.com/cd/B19306_01/appdev.102/b14258/u_file.htm

      - You will need to do a little research and testing to get this feature figured out and working.

      - Stored programs that you may need would include:

        - UTL_FILE.FOPEN
        - UTL_FILE.PUT
        - UTL_FILE.NEW_LINE
        - UTL_FILE.FCLOSE

Before starting your application coding, make sure to:

1. Ensure that you have created the WKIS database tables successfully. These files are located in the Course Resource area of D2L.

2. Design and write a PL/SQL program using the information outlined above.
    a. Do not worry about invalid data going into the database from the delimited text file that was processed (other than as noted with the status flag in your trigger). Data errors will be handled when the transactions in the NEW_TRANSACTIONS table are processed (this is the process you created in Assignment 1).
    b. There should be no hard coded values in your Java code (i.e. user name, path and file names). All this information should be prompted from the user.

3. When calling stored functions from your Java code, do **not** call the function through a SELECT statement, callable statements must be used.

4. There should be no hard coding of data except where specifically indicated to do so in the assignment problem.

Evaluation

Criteria	Completed
1. GUI (3)	
2. Database connection. (2)	
3. Function to check if user has correct permissions. (3)	
4. Java calls function to check user permissions. (2)	
5. SQL*Loader called from Java with data loaded into PAYROLL_LOAD. (4)	
6. Dynamic generation of the SQL*Loader control file. (2)	
7. Trigger to process PAYROLL_LOAD data. (8)	
8. Procedure that performs all month end tasks. (8)	
9. Java calls to procedure to perform month end tasks. (2)	
10. Procedure to export data. (4)	
11. Java calls database to create Oracle directory "alias". (2)	
12. Java calls procedure to export data. (2)	