

CPRG 307

Assignment 1 (Modules 1 – 5)

Student: _____

Mark: _____ / 9

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_A1.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_A1.zip

Problem

This assignment is based on the We Keep It Storage (WKIS) Company's system, an accounting system. Transactions (this is a double-entry accounting system) will be taken from a holding table (NEW_TRANSACTIONS) and inserted into the TRANSACTION_DETAIL and TRANSACTION_HISTORY tables. At the same time, the appropriate account balance will be updated in the ACCOUNT table. This system follows a double entry accounting system with the accounting rules presented in class.

1. Ensure that you have created the WKIS database tables successfully. These files are located in the Course Resource area of D2L.
2. Design, write a PL/SQL program, and thoroughly test the coded solution using the information outlined above and the guidelines / restrictions below.
 - a. You can assume that every transaction number is unique (there will be no duplicates) for each "transaction". Remember that a transaction is a unit – will be made up of more than one row. All rows that represent a single transaction will have the same transactional history information.
 - b. Using two cursors would make this problem easier. See Module 4 slides and the Module 4 exercise for an example.
 - c. As long as the debits equal the credits in each transaction, you can assume that the accounting equation for each transaction holds true.
 - d. After a transaction has been successfully processed, remove it from the NEW_TRANSACTIONS table. Transactions that produce an error should remain in the NEW_TRANSACTIONS table.
 - e. An error in one transaction should not prevent the processing of other transactions (i.e. make sure you do not leave the looping structure when an error occurs).
 - f. Only the first error in a transaction should be recorded in the error log table (i.e. a specific transaction number should only appear once in the error log table). If the error is a missing transaction number, a single entry can be recorded in the error log table for all rows missing a transaction number or can have an entry recorded for each row missing a transaction number.
 - g. All required tables for this assignment, including the error log, are created with the provided script. Do not create any additional tables or modify the existing tables.

- h. Do **not** use a table of records, or any other type of array, in your solution to this problem.
 - i. SELECT INTO cannot be performed against the NEW_TRANSACTIONS table. NEW_TRANSACTIONS can only be referenced by an explicit cursor (use your cursor for any needed values from this table).
 - j. The solution must be done with **one** anonymous block (multiple embedded blocks are fine as these are not considered separate anonymous blocks).
 - k. Stored programs **cannot** be used.
 - l. **Do not use** GOTOs, EXITs, or SAVEPOINTS. CONTINUEs can be used if done appropriately (do not use as you would a *break* in Java).
 - m. There should be **no** hard coding of values anywhere in your code except for the transaction type ('C', 'D') which should be hard coded only in the DECLARE section.
3. Your program must handle all exceptions and write the transactional history information that caused the error as well as the error message to the WKIS_ERROR_LOG table. Error messages must be descriptive. Specific errors to catch:
- a. Missing transaction number (NULL transaction number)
 - b. Debits and credits are not equal
 - c. Invalid account number
 - d. Negative value given for a transaction amount
 - e. Invalid transaction type
- Your code should handle unanticipated errors as well (anything outside of those listed above). The error generated would be the system generated error message as we cannot provide a customized descriptive error for this type of problem. For our purposes, testing does not need to be performed for this type of error.
 - The data provided in NEW_TRANSACTIONS is clean (should not cause any errors). To test your error handling in the code, you will need to modify the transactional data as needed to set up those tests appropriately.

Submission

This assignment will be graded based on test results when executing your coded solution. The Test Feedback / Marking Guide is below. You have two opportunities to submit this assignment.

Submission #1

- Create your coded solution expecting only clean data will be used by the instructor to test (do not worry about error handling on this submission).
- Submit your coded solution to D2L by the due date / time indicated for *Submission #1*.
- Your instructor, within one week of the submission date, will test your code using their test data and the Test Feedback form, assign a grade based on the results of these tests, and will post your grade along with a copy of the feedback form to D2L.
- This will be your assignment mark if you do nothing further.
 - If the assignment was not submitted, the current grade will be zero.
- This first submission to D2L will only allow you one submission attempt. Your instructor may start marking your submission before the deadline to provide you quicker feedback to make corrections for your second submission.

Submission #2

- Add error handling to your code.
- Take the feedback supplied from the first submission and make modifications as needed.
- Submit your updated coded solution to D2L by the due date / time indicated for *Submission #2*.
 - This is the final submission opportunity for this assignment.
 - You can submit even if you did not submit on the first submission attempt.
- Your instructor will use the same Testing Feedback / Marking Guide to evaluate the second submission.
- The higher mark between the two submissions will be your grade on this assignment.
- Unlike the first submission, with the second submission you can submit the assignment as many times to D2L as you need up to the deadline. Your instructor will always evaluate the last submission attempt unless you direct them otherwise.

Student: _____

Mark: ____ / 9

Test Type	Complete	Partially Complete	Incomplete (0 Final Score on Assignment)	Score
(Automatic zero mark)			<input type="checkbox"/> Cannot test application as syntax errors exist <input type="checkbox"/> Runtime error is occurring preventing complete testing of the application <input type="checkbox"/> Database structure has been modified resulting in syntax or runtime errors <input type="checkbox"/> Data changes are not saved correctly (no COMMIT or incomplete COMMIT)	

Test Type	Complete (4 Marks)	Partially Complete (2.0 – 3.5 Marks)	Incomplete (0 – 3 Marks)	Score (4)
Dynamic Black-Box • Test-to-pass	<input type="checkbox"/> All DML successfully performed assuming clean data	<input type="checkbox"/> Not all transactions successfully processed into TRANSACTION_HISTORY <input type="checkbox"/> Not all transactions successfully processed into TRANSACTION_DETAIL <input type="checkbox"/> Not all transactions successfully removed from NEW_TRANSACTIONS <input type="checkbox"/> Not all account balances successfully updated	<input type="checkbox"/> No transactions successfully processed into TRANSACTION_HISTORY <input type="checkbox"/> No transactions successfully processed into TRANSACTION_DETAIL <input type="checkbox"/> No transactions successfully removed from NEW_TRANSACTIONS <input type="checkbox"/> No account balances successfully updated	

Test Type	Complete (5 Marks)	Partially Complete (2.5 – 4.5 Marks)	Incomplete (0 - 4 Marks)	Score (5)
Dynamic Black-Box • Test-to-fail	<input type="checkbox"/> All requested errors successfully recorded and those errored transactions are not processed into the database tables	<input type="checkbox"/> Some error messages are inaccurate and / or not descriptive to the actual error <input type="checkbox"/> Not all requested errors are be caught in a controlled manner to allow for a descriptive error message (I.E. get a standard database constraint error message) <input type="checkbox"/> Not all requested errors are being caught <input type="checkbox"/> Some transactions still being processed even though in error <input type="checkbox"/> Some transactions in error are being removed from the NEW_TRANSACTIONS table	<input type="checkbox"/> All error messages are inaccurate and / or not descriptive to the actual error <input type="checkbox"/> All requested errors are not being caught in a controlled manner to allow for a descriptive error message (I.E. get a standard database constraint error message) <input type="checkbox"/> No requested errors are being caught <input type="checkbox"/> All transactions still being processed even though in error <input type="checkbox"/> All transactions in error are being removed from the NEW_TRANSACTIONS table	



Test Type	Complete (No Deductions)	Partially Complete	Incomplete (Deduction: 0 – 3 Marks)	Score
Static White-Box	<input type="checkbox"/> All coding restriction guidelines being followed		<input type="checkbox"/> Data values are hard coded in the application (other than values as outlined in the assignment) <input type="checkbox"/> GOTO, EXIT, SAVEPOINT or arrays (table of records / collection) used <input type="checkbox"/> SELECT INTO on NEW_TRANSACTIONS being performed <input type="checkbox"/> Multiple anonymous blocks used (embedded blocks are different than anonymous blocks) <input type="checkbox"/> Stored programs included <input type="checkbox"/> Documentation (header and inline comments) not present	

