CPRG 307
Lab 4: Control Structures

Student: _____

Mark: ___ / 2 (*one mark for each deliverable*)

Lab Objectives

Here is what you will be able to do when you complete each objective:

- Explain how various forms of the IF statement control program flow.
- Create various looping structures to control program flow.
- Explain the purpose of cursors in a program.
- Create cursors to control the retrieval of data.
- Create cursor FOR loops.
- Identify the different between basic cursor loops and cursor FOR loops.
- Use the CASE structure to replace complex IF statements.

Lab Instructions

To complete this lab, follow the steps below. This lab is due on the day and time indicated by your instructor.

Steps:

☐ 1. ATTEND the lecture on the material that will be performed in the lab exercise.

☐ 2. COMPLETE the out-of-class learning activities as indicated by your instructor.

☐ 3. COMPLETE the prelab tasks identified in the lab document before the lab class, making sure to submit solutions to the appropriate forum and thread in the D2L discussion board.

☐ 4. COMPLETE the tasks identified in the lab document, making sure to submit solutions to the appropriate forum and thread in the D2L discussion board.

☐ 5. COMPLETE the post lab tasks identified in the lab document after the lab has been completed.

Deliverables

☐ 1. SUBMIT the complete and tested prelab code by the date and time indicated by your instructor to the appropriate forum and topic in the D2L discussion board.

☐ 2. SUBMIT the complete and tested lab code by the date and time indicated by your instructor to the appropriate forum and topic in the D2L discussion board.

For this lab, all code should be placed in the body of the discussion board posts – <u>not</u> as an attachment.

Unless stated otherwise, code from <u>all</u> tasks should be included in discussion board posts.

Prelab Tasks

The following questions use the Astra Talent Agency (ATA) table set.

☐ 1. Write the PL/SQL code for the following problem:

Add a new column to the ATA_ENTERTAINER table called MORE_THAN_ONE with a NUMBER datatype (this code will be outside of your PL/SQL code). This column will hold the number of style types an entertainer has if they have more than one (if they have one, place a NULL in this column).

Your code should, using an explicit cursor and loop structure, go through each entertainer and determine the number of styles they have. If an entertainer has more than one, modify the value in the MORE_THAN_ONE column for that entertainer with the number of styles (a NULL should be put in this column if not more than one). Use FOR UPDATE and WHERE CURRENT OF as part of your solution. Use a basic loop to solve this problem, and an IF for any decision structures.

☐ 2. Write the PL/SQL code for the following problem:

Solve the same problem as Task 1 in this lab but use a cursor FOR loop instead of a basic loop, a searched CASE instead of an IF for decision structures and do not use a FOR UPDATE and WHERE CURRENT OF in your solution.

Lab Tasks

The following questions use the EMP table set.

Although you do not have to submit a flowchart or test plan for this lab problem, it is recommended you still create them as this will help you to breakdown and solve the problem. When your instructor posts the solutions for the lab to the discussion board, they will also include the flowchart, test plan, and test code to provide you with examples of the entire problem lifecycle.

☐ 1. CREATE **and** thoroughly test a PL/SQL coded solution for the following problem:

The president of the company wants to ensure that everyone is receiving a "fair" wage. He has asked you to modify salaries of everyone, except him, using the following guidelines (each point uses the salary from the previous step unless otherwise specified):

- If anyone makes more than the president does, they should have their salaries reduced by 50% or reduced to 25% less than the president makes (whichever one would be less).
- If anyone makes less than $100, their salary should be increased by 10%, but only if the **original** average salary for the entire company is still more than their new raised salary.
  - The company has decided that they only want to look at the average salary **before** the changes from the first step are performed rather than after.
- If an employee's commission is more than 22% of their salary, the commission should be changed to the lowest commission in their department (excluding anyone who has a commission with a 0 value).
  - If an employee does not have a commission (i.e. NULL or 0 value), no changes should be made to their commission.

Restrictions:

- Must use one looping structure (with cursor)
- Can only use a SELECT…INTO to get: (1) the average salary of the company, (2) the president's salary, and (3) the lowest commission in a department
  - The first two queries identified above must be performed outside of the looping structure
  - All DML commands must be inside the loop and utilize the cursor data
  - The third query mentioned above must be inside the loop and utilize the cursor data
  - Only **one** DML command can be used in the coded solution
- Can only hard code information that was provided in the problem
  - 'PRESIDENT'

- 50%
- 25%
- $100
- 10%
- 22%

- Hard coded values should be defined as constants and then the constants used in the body of the code
- Can assume the company has only one president

Extra Questions to Practice

Below are extra questions to practice for the material in this module (these are extra to the lab and are for practice only):

☐ 1. Complete the following textbook problem: *Assignment 4-1: Using an explicit cursor* on page 167 of your textbook. If you have questions on the results of this problem, please call your instructor over.

☐ 2. Complete the following textbook problem: *Assignment 4-2: Using a CURSOR FOR loop* on page 167-168 of your textbook. If you have questions on the results of this problem, please call your instructor over.

☐ 3. CREATE the PL/SQL code for the flowchart created for the textbook problem: *Assignment 2-7: Creating a flowchart* on page 83 of your textbook. Your code should output to the screen the shipping cost accordingly.

Use the following tests:

- 2, non-member
- 3, member
- 12, member
- 6, non-member
- 7, member

Post Lab Tasks

☐ 1. COMPARE your posted solutions to those posted by your instructor. If you are unsure why there are differences between the solutions, make sure to talk to your instructor.