Assignment #4 (CSC452 &352),

Due date 4/27, 8:00PM (total points = 100)

Please submit one .txt file. Such as:

Q1

1. T

2. T

3. T

4. T

5. T

6. T

7. T

or 1, 3, 5, 7 are T

2, 4, 6 are F

Q2.(1) a

Q2 (2) a

REM Q3.

DECLARE

Maxno number := 119;

counter number := 100;

= = = =

Q1. True or False (20 points)

1. In a PL/SQL program, an exception declared in the outer block is global, it is valid for its inner block.
2. The exception name declared in the inner block is global for its outer block.
3. Most of the system internally defined errors do not have names, thus program cannot write the right handler to catch that individual exception (besides using When others).
4. A user-defined exception always has a message (description) when defined.
5. User may associate a name to a system internally defined error (that has no name yet), thus programmer can handle this error in the exception section.
6. The command of “PRAGMA EXCEPTION\_INIT (*exception\_name*, *error\_code*)”

must be placed in the declarative section, and after the declaration of that

1. In the Exception section, there may have several exception handlers.

Q2. Multiple choices (10 points each).

Choose one and only one which is not true, or you think the least close to correct.

(1)

1. The system will automatically raise a system defined exception once it occurs in a program, no matter it has name or not.
2. All the system defined exceptions have a message.
3. All the system defined exceptions have a code (number).
4. None of any system defined exceptions can be raised by program.

(2)

1. The exception handlers in the exception section can only handle the exceptions that occur in the executable section in the same block.
2. In a PL/SQL program, if an exception occurs in the execution section in an inner block, and it is handled by its Exception section in that inner block, then control will be passed to the next statement in its outer (enclosing) block. Just same as there is no error occrued.
3. In a PL/SQL program, if an exception occurs in the execution section in the inner block, and it is not handled by its Exception section in that inner block, then this exception will propagate to the exception section in its outer (enclosing) block.
4. In a PL/SQL program, if an exception occurs in the execution section in the inner block, and it is not handled by its Exception section in that inner block, then the control will be passed to the next statement in its outer (enclosing) block.

Q3. (20 points) Write a simple anonymous block.

In this block, you declare a user-defined exception named Exceed\_Limit. In the block, you will run a loop with a counter starting from 1, the program will increase the counter by 2 each time with each run in a loop. The program will raise the exception Exceed\_Limit when the counter is greater than 10. Your program will have an exception section that will trap this exception and print out an error message.

Note, as routine, the program needs to define the exception in the declaration section, raise the exception in the executable section, and handle the exception in the exception section.

Q4. (20 points) The company makes a new rule that the minimal salary for employees with job ID as IT\_PROG (IT Programmer) will be $5,000. Your program will define a cursor to retrieve the employee’s ID and salary for those employees whose job ID as IT\_PROG.

Your program will check the amount of the salary of these employees; if that amount is equal or greater than $5,000, then it is fine; if not, your program will raise an exception defined as Sal\_Below\_Min. In the exception handler, your program will print out a message with the info of the employee ID and the current salary amount.

As a basic request of this question, you are required to find one violation, that is good enough. That means, once your program find one violation then the program raises the exception, print out that employee’s info, then this block is done.

If you like some challenging, then your program may use the technique "continue past exception", to print all the IT programmers that their salaries are lower than limit.

Q5. (20 points)One of the popular ways in exception handling is to make a record of the error into a table for future review. The code below will create a table to store the error information,

DROP TABLE log\_error; -- in case you have that table created, otherwise ignore this.

CREATE TABLE log\_error (

Occur\_date DATE DEFAULT SYSDATE,

Username VARCHAR2 (15) DEFAULT USER,

Err\_code NUMBER,

Err\_msg VARCHAR2 (255));

\*\* Note, we can use "SYSDATE" for Occur\_date, "USER" for username.

If your tables are created by running the script provided on D2L, then the manager\_id in the employees table is a foreign key referencing to employee\_id in the same table employees. Thus user cannot simply delete an employee if that employee is also a manager (his ID being on the list of manager\_id in employees table).

Write a PL/SQL block that will run two SQL delete commands, one to delete an employee with ID 104; second time, delete the employee with ID 123. To keep the data as were created, it is recommended to “rollback” these deletions.

In the executable section, you will define an exception section with OTHERS handler, that will catch up any unexpected error. In that OTHERS handler, your program will use SQLCODE and SQLERRM to catch the error number and error message. Your program will execute an insert command to add a record into the log\_error table when an error occurs.

After you run the PL/SQL block, you need to run a SQL statement separately to display the contents of the log\_error table, that should include the record just inserted when an error was caught in the last program.