PL*SQL

Exercise 7

1. Write a PL*SQL block that prompts the user to enter the salary of an employee. Your program should display the name of the employee (from the EMP table) who's getting that salary. If more than 1 employee is receiving that salary, or if no employees exist getting that salary, your program should display appropriate messages. Use *too_many_rows* and *no_data_found* exceptions to achieve this. Display the results on the screen using dbms_output.put_line.

2. Write a PL*SQL block to check if any employee from EMP table is receiving a salary greater than 9999.99. Make the use of *value_error* exception to achieve this. Display the results on the screen using dbms_output.put_line.

3. Create a user-defined exception by the name of *exp_check*. Select the ename and hiredate of all employees into a cursor. Your program should calculate the experience of all the employees in years, and insert the ename and experience of each employee into tempp table. If any employee has experience less than 2 years, the program should be aborted with a suitable message. Raise the user-defined exception *exp_check* to achieve this. Display the results on the screen using dbms_output.put_line.

4. Write a PL*SQL function to take three parameters, the sides of a triangle. The sides of the triangle should be accepted from the user. The function should return a Boolean value:- *true* if the triangle is valid, *false* otherwise. A triangle is valid if the length of each side is less than the sum of the lengths of the other two sides. Check if the dimensions entered by the user can form a valid triangle. Display the results on the screen using dbms_output.put_line.

5. Write a function that generates a random number between 1 and 10. Use any logic of your choice to achieve this. Display the results on the screen using dbms_output.put_line.

6. Design a structure to store length in yards, feet, and inches (for example, 7 yards, 2 feet, 3 inches). Your program should accept 2 length measurements from the user. Write a PL*SQL procedure to find the difference between two measurements as represented by these structures. Display the results on the screen using dbms_output.put_line.

7. Create a function that accepts a string of *n* characters and exchanges the first character with the last, the second with the next – to – last, and so forth until *n* exchanges have been made. What will the final string look like? Write the function to verify your conclusion. Display the results on the screen using dbms_output.put_line.