Exercise 8 Exception handling

Write a PL/SQL block to illustrate the programmer-defined exceptions. Get the salary of an employee and check it with the job's salary range. If the salary is below the range, raise exception BELOW_SALARY_RANGE. If the salary is above the range, raise exception ABOVE_SALARY_RANGE.

Objective: The objective of this exercise is to enable you to make use of exception handling in PL/SQL.

Procedure and description:

The EXCEPTION section is an optional part of any PL/SQL block. If this section is omitted and errors are encountered, the block will be terminated. Some errors that are encountered may not justify the immediate termination of a block, so the EXCEPTION section can be used to handle specified errors or user-defined exceptions in an orderly manner. Exceptions can be user-defined.

Exceptions are raised in a block by using the command RAISE. Exceptions can be raised explicitly by the programmer, whereas internal database errors are automatically, or implicitly, raised by the database server.

The General Syntax for coding the exception section

```
 DECLARE
   Declaration section
 BEGIN
   Exception section
 EXCEPTION
 WHEN ex_name1 THEN
    -Error handling statements
 WHEN ex_name2 THEN
    -Error handling statements
 WHEN Others THEN
   -Error handling statements
END;
```

Example Exception code:

```
SET SERVEROUTPUT ON SIZE 100000;
    DECLARE
      -- define exceptions
      BELOW_SALARY_RANGE EXCEPTION;
      ABOVE_SALARY_RANGE EXCEPTION;
      -- salary variables
      n_salary employees.salary%TYPE;
      n_min_salary employees.salary%TYPE;
      n_max_salary employees.salary%TYPE;
      -- input employee id
      n_emp_id employees.employee_id%TYPE := &emp_id;
    BEGIN
      SELECT salary,
             min_salary,
             max_salary
      INTO n_salary,
           n_min_salary,
           n_max_salary
      FROM employees
      INNER JOIN jobs ON jobs.job_id = employees.job_id
      WHERE employee_id = n_emp_id;

      IF n_salary < n_min_salary THEN
        RAISE BELOW_SALARY_RANGE;
      ELSIF n_salary > n_max_salary THEN
         RAISE ABOVE_SALARY_RANGE;
      END IF;

      DBMS_OUTPUT.put_line('Employee ' || n_emp_id ||
                           ' has salary $' || n_salary );

     EXCEPTION
      WHEN BELOW_SALARY_RANGE THEN
        DBMS_OUTPUT.put_line('Employee ' || n_emp_id ||
                                  ' has salary below the salary range');
       WHEN ABOVE_SALARY_RANGE THEN
         DBMS_OUTPUT.put_line('Employee ' || n_emp_id ||
                                  ' has salary above the salary range');
       WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE('Employee ' || n_emp_id || ' not found');
    END;
    /
```

Algorithm: The steps for this exercise are given below:

Step – 1: Write the PL/SQL code to create table that accept employee ID from user and display employee.

Step – 2: include the condition for the exception raised.

Step – 3: Execute.

Expected Output:

PL/SQL exception, raise and handle it in exception handler section of PL/SQL block.