PLSQL ASSIGNMENT 1 FOR ALL GROUPS

INSTRUCTIONS:

1) Deadline: Friday, 23/February/2024 at 10:00 am

2) Submission: Google classroom, code: s2r54l3 (To join with the link: https://classroom.google.com/c/NjY0MDIyNDgwNzQ1?cjc=s2r54l3)

3) File naming: studentcode_fullName_GroupDay (eg: 1111_HuguetteSandrine_Monday)

4) File content: Question number, codes, and screenshots

5) Assignment Format: pdf

6) Submission MUST be made on the google classroom ONLY

QUESTIONS:

1) Create a sample table, add 5 records to it.(/5 Marks)

Use the created table in (1) to answer the below questions:

2) Write a PLSQL program that raises and catches the row type mismatch exception (/5 Marks)

3) Write a PLSQL program that raises the case not found exception using the simple case expression (/5 Marks)

4) Write a PLSQL program that illustrates how to raise a user defined exception associated with an error code (/5 Marks)

5) Write a PLSQL program illustrate the use of nested case searched case statement inside a simple case statement (/5 Marks)

Task 1: Create a sample table and add 5 records:

-- Create a sample table

```
CREATE TABLE sample_table (
   id NUMBER,
   name VARCHAR2(50),
   age NUMBER
);

-- Add 5 records to the table
INSERT INTO sample_table VALUES (1, 'John', 25);
INSERT INTO sample_table VALUES (2, 'Jane', 30);
INSERT INTO sample_table VALUES (3, 'Alice', 22);
INSERT INTO sample_table VALUES (4, 'Bob', 28);
INSERT INTO sample_table VALUES (5, 'Eve', 35);
```

Task 2: Write a PL/SQL program that raises and catches the row type mismatch exception:

```
DECLARE
   -- Declare a user-defined exception for row type mismatch
   row_type_mismatch EXCEPTION;

   -- Declare a variable with the table type
   TYPE SampleTableType IS TABLE OF sample_table%ROWTYPE;

   -- Declare a variable of the table type
   sample_table_data SampleTableType;

BEGIN
   -- Assign a value to the variable (this could be a result of a query)
   sample_table_data := SampleTableType(1, 'John', 25);
```

```
  -- Try to insert the row into the sample_table, which will raise an exception
  INSERT INTO sample_table VALUES sample_table_data;

EXCEPTION
  -- Catch the row type mismatch exception
  WHEN row_type_mismatch THEN
    DBMS_OUTPUT.PUT_LINE('Row type mismatch exception caught!');
END;
```

Task 3: Write a PL/SQL program that raises the case not found exception using the simple case expression

```
DECLARE
  -- Declare a variable
  my_variable VARCHAR2(10) := 'XYZ';

BEGIN
  -- Use a simple case statement
  CASE my_variable
    WHEN 'ABC' THEN
      DBMS_OUTPUT.PUT_LINE('Case found: ABC');
    WHEN '123' THEN
      DBMS_OUTPUT.PUT_LINE('Case found: 123');
    -- Raise the case not found exception
    WHEN OTHERS THEN
      RAISE CASE_NOT_FOUND;
  END CASE;

EXCEPTION
```

```
   -- Catch the case not found exception
   WHEN CASE_NOT_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('Case not found exception caught!');
END;
```

Task 4: Write a PL/SQL program that illustrates how to raise a user-defined exception associated with an error code

```
DECLARE
   -- Declare a user-defined exception with an error code
   my_exception EXCEPTION;
   PRAGMA EXCEPTION_INIT(my_exception, -20001);

BEGIN
   -- Raise the user-defined exception with an error code
   RAISE my_exception;

EXCEPTION
   -- Catch the user-defined exception and display the error code
   WHEN my_exception THEN
      DBMS_OUTPUT.PUT_LINE('User-defined exception caught with error code: ' || SQLCODE);
END;
```

Task 5: Write a PL/SQL program to illustrate the use of nested case searched case statement inside a simple case statement

```
DECLARE
   -- Declare variables
   x NUMBER := 10;
```

```
   y NUMBER := 5;


BEGIN
   -- Use a simple case statement
   CASE
      WHEN x > y THEN
         -- Use a nested searched case statement
         CASE
            WHEN x > 0 THEN
               DBMS_OUTPUT.PUT_LINE('x is greater than y and positive');
            WHEN x < 0 THEN
               DBMS_OUTPUT.PUT_LINE('x is greater than y and negative');
         END CASE;
      WHEN x < y THEN
         DBMS_OUTPUT.PUT_LINE('x is less than y');
      ELSE
         DBMS_OUTPUT.PUT_LINE('x is equal to y');
   END CASE;
END;
```