NAME: SHREYA BANSAL
PRN: 23070521144
BATCH:B2

Practical 07

Write and execute PL/SQL function to print /return binary equivalent of decimal number.

Introduction

A PL/SQL function is a subprogram that computes and returns a value. It helps in reusability, modular programming, and efficient database operations.

Key Concepts Used in This Program

- Functions in PL/SQL: A function must have a return type and return a value.
- Loops in PL/SQL: We use loops to repeatedly divide the decimal number by 2 to obtain its binary equivalent.
- String Operations: We build the binary number as a string.

PL/SQL Function to Convert Decimal to Binary

Steps to Convert Decimal to Binary in PL/SQL

1. Take a decimal number as input.
2. Use a LOOP to repeatedly divide the number by 2.
3. Store the remainders (0 or 1) in reverse order.
4. Return the final binary string.

PL/SQL Function Code

```
CREATE OR REPLACE FUNCTION decimal_to_binary(n IN NUMBER) RETURN
VARCHAR2 IS
    binary_result VARCHAR2(100) := ''; -- Variable to store the
binary equivalent
    num NUMBER := n; -- Copy of the input number
    remainder NUMBER; -- Stores remainder after division
BEGIN
    -- Check for zero case
    IF num = 0 THEN
        RETURN '0';
    END IF;

    -- Loop to convert decimal to binary
    WHILE num > 0 LOOP
        remainder := MOD(num, 2); -- Get remainder when divided
by 2
        binary_result := remainder || binary_result; -- Build
binary string in reverse
        num := TRUNC(num / 2); -- Reduce number by dividing by
2
    END LOOP;

    RETURN binary_result; -- Return final binary value
END decimal_to_binary;
/
```

How to Execute the Function

Call the Function Using PL/SQL Block

```
DECLARE
```

```
    decimal_num NUMBER := 10; -- Example decimal number
    binary_value VARCHAR2(100);
BEGIN
    binary_value := decimal_to_binary(decimal_num);
DBMS_OUTPUT.PUT_LINE('Binary equivalent of ' || decimal_num ||
' is: ' || binary_value);
END;
/
```

Expected Output:

```
Binary equivalent of 10 is: 1010
```

Explanation of the Code

Step	Description
Function Creation	Defines `decimal_to_binary` function with input `n` (decimal number).
Binary Result Variable	Stores the binary representation as a string.
Loop Execution	Repeatedly divides `num` by 2, storing remainders.
String Concatenation	Builds binary number in reverse order.
Return Statement	Returns the final binary string.

Task

1. Modify the function to display step-by-step conversion while calculating binary.

```
SQL> CREATE OR REPLACE FUNCTION decimal_to_binary(num IN NUMBER) RETURN VARCHAR2 IS
  2      remainder NUMBER;
  3      binary_result VARCHAR2(100) := '';
  4      num_copy NUMBER;
  5  BEGIN
  6      num_copy := num;
  7
  8      IF num_copy = 0 THEN
  9          DBMS_OUTPUT.PUT_LINE('Step 0: 0 / 2 = 0, Remainder = 0');
 10          RETURN '0';
 11      END IF;
 12
 13      DBMS_OUTPUT.PUT_LINE('Converting ' || num || ' to binary:');
 14
 15      WHILE num_copy > 0 LOOP
 16          remainder := MOD(num_copy, 2);
 17          binary_result := remainder || binary_result;
 18          DBMS_OUTPUT.PUT_LINE('Step: ' || num_copy || ' / 2 = ' || TRUNC(num_copy / 2) ||
 19                               ', Remainder = ' || remainder);
 20          num_copy := TRUNC(num_copy / 2);
 21      END LOOP;
 22
 23      RETURN binary_result;
 24  END decimal_to_binary;
 25  /

Function created.

SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2      decimal_num NUMBER := 10;
  3      binary_value VARCHAR2(100);
  4  BEGIN
  5      binary_value := decimal_to_binary(decimal_num);
  6      DBMS_OUTPUT.PUT_LINE('Binary equivalent of ' || decimal_num || ' is: ' || binary_value);
  7  END;
  8  /
Converting 10 to binary:
Step: 10 / 2 = 5, Remainder = 0
Step: 5 / 2 = 2, Remainder = 1
Step: 2 / 2 = 1, Remainder = 0
Step: 1 / 2 = 0, Remainder = 1
Binary equivalent of 10 is: 1010

PL/SQL procedure successfully completed.
```

2. Write a PL/SQL block to accept user input for the decimal number and call the function.

```
SQL> SET SERVEROUTPUT ON;
SQL>
SQL> DECLARE
  2      decimal_num NUMBER;
  3      binary_value VARCHAR2(100);
  4  BEGIN
  5      decimal_num := &Enter_Decimal_Number;
  6
  7      binary_value := decimal_to_binary(decimal_num);
  8
  9      DBMS_OUTPUT.PUT_LINE('Binary equivalent of ' || decimal_num || ' is: ' || binary_value);
 10  END;
 11  /
Enter value for enter_decimal_number: 10
old   5:      decimal_num := &Enter_Decimal_Number;
new   5:      decimal_num := 10;
Converting 10 to binary:
Step: 10 / 2 = 5, Remainder = 0
Step: 5 / 2 = 2, Remainder = 1
Step: 2 / 2 = 1, Remainder = 0
Step: 1 / 2 = 0, Remainder = 1
Binary equivalent of 10 is: 1010

PL/SQL procedure successfully completed.
```

3. Modify the function to store binary values in a table (`binary_conversions`).

```
SQL> CREATE TABLE binary_conversions (
  2      decimal_number NUMBER PRIMARY KEY,
  3      binary_value VARCHAR2(100)
  4  );

Table created.

SQL> CREATE OR REPLACE FUNCTION decimal_to_binary(num IN NUMBER) RETURN VARCHAR2 IS
  2      remainder NUMBER;
  3      binary_result VARCHAR2(100) := '';
  4      num_copy NUMBER;
  5  BEGIN
  6      num_copy := num;
  7
  8      IF num_copy = 0 THEN
  9          DBMS_OUTPUT.PUT_LINE('Step 0: 0 / 2 = 0, Remainder = 0');
 10          INSERT INTO binary_conversions VALUES (num, '0');
 11          RETURN '0';
 12      END IF;
 13
 14      DBMS_OUTPUT.PUT_LINE('Converting ' || num || ' to binary:');
 15
 16      WHILE num_copy > 0 LOOP
 17          remainder := MOD(num_copy, 2);
 18          binary_result := remainder || binary_result;
 19          DBMS_OUTPUT.PUT_LINE('Step: ' || num_copy || ' / 2 = ' || TRUNC(num_copy / 2) ||
 20                               ', Remainder = ' || remainder);
 21          num_copy := TRUNC(num_copy / 2);
 22      END LOOP;
 23
 24      INSERT INTO binary_conversions VALUES (num, binary_result);
 25      COMMIT;
 26
 27      RETURN binary_result;
 28  END decimal_to_binary;
 29  /

Function created.

SQL> SET SERVEROUTPUT ON;
SQL>
SQL> DECLARE
  2      decimal_num NUMBER;
  3      binary_value VARCHAR2(100);
  4  BEGIN
  5      decimal_num := &Enter_Decimal_Number;
  6
  7      binary_value := decimal_to_binary(decimal_num);
  8
  9      DBMS_OUTPUT.PUT_LINE('Binary equivalent of ' || decimal_num || ' is: ' || binary_value);
 10  END;
 11  /
Enter value for enter_decimal_number: 10
old   5:     decimal_num := &Enter_Decimal_Number;
new   5:     decimal_num := 10;
Converting 10 to binary:
Step: 10 / 2 = 5, Remainder = 0
Step: 5 / 2 = 2, Remainder = 1
Step: 2 / 2 = 1, Remainder = 0
Step: 1 / 2 = 0, Remainder = 1
Binary equivalent of 10 is: 1010

PL/SQL procedure successfully completed.
```