Practical 07

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

SHIVANSH LOHANI

23070521141

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(n IN NUMBER) RETURN VARCHAR2 IS
  2      binary_result VARCHAR2(100) := '';
  3      num NUMBER := n;
  4      remainder NUMBER;
  5  BEGIN
  6      IF num = 0 THEN
  7          DBMS_OUTPUT.PUT_LINE('0');
  8          RETURN '0';
  9      END IF;
 10
 11      DBMS_OUTPUT.PUT_LINE('Step-by-step conversion:');
 12
 13      WHILE num > 0 LOOP
 14          remainder := MOD(num, 2);
 15          binary_result := remainder || binary_result;
 16          DBMS_OUTPUT.PUT_LINE('Decimal: ' || num || ' -> Remainder: ' || remainder);
 17          num := TRUNC(num / 2);
 18      END LOOP;
 19
 20      RETURN binary_result;
 21  END decimal_to_binary;
 22  /

Function created.
```

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

```
SQL> SET SERVEROUTPUT ON;
SQL> DECLARE
  2      decimal_num NUMBER;
  3      binary_value VARCHAR2(100);
  4  BEGIN
  5      decimal_num := &Enter_Decimal_Number;
  6      binary_value := decimal_to_binary(decimal_num);
  7      DBMS_OUTPUT.PUT_LINE('Binary equivalent of ' || decimal_num || ' is: ' || binary_value);
  8  END;
  9  /
Enter value for enter_decimal_number: 12
old   5:     decimal_num := &Enter_Decimal_Number;
new   5:     decimal_num := 12;
Step-by-step conversion:
Decimal: 12 -> Remainder: 0
Decimal: 6 -> Remainder: 0
Decimal: 3 -> Remainder: 1
Decimal: 1 -> Remainder: 1
Binary equivalent of 12 is: 1100

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(n IN NUMBER) RETURN VARCHAR2 IS
  2      binary_result VARCHAR2(100) := '';
  3      num NUMBER := n;
  4      remainder NUMBER;
  5  BEGIN
  6      IF num = 0 THEN
  7          INSERT INTO binary_conversions VALUES (0, '0');
  8          RETURN '0';
  9      END IF;
 10
 11      WHILE num > 0 LOOP
 12          remainder := MOD(num, 2);
 13          binary_result := remainder || binary_result;
 14          num := TRUNC(num / 2);
 15      END LOOP;
 16
 17      INSERT INTO binary_conversions VALUES (n, binary_result);
 18
 19      RETURN binary_result;
 20  END decimal_to_binary;
 21  /

Function created.

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  /
Binary equivalent of 10 is: 1010

PL/SQL procedure successfully completed.

SQL>
SQL> SELECT * FROM binary_conversions;

DECIMAL_NUMBER
--------------
BINARY_VALUE
--------------------------------------------------------------------------------
           10
1010
```