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.

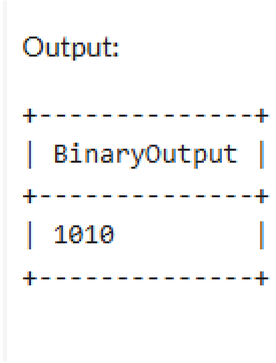
PL/SQL Function Code

```
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;
```



Output:

```
+--------------+
| BinaryOutput |
+--------------+
| 1010         |
+--------------+
```

```
/
```

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:

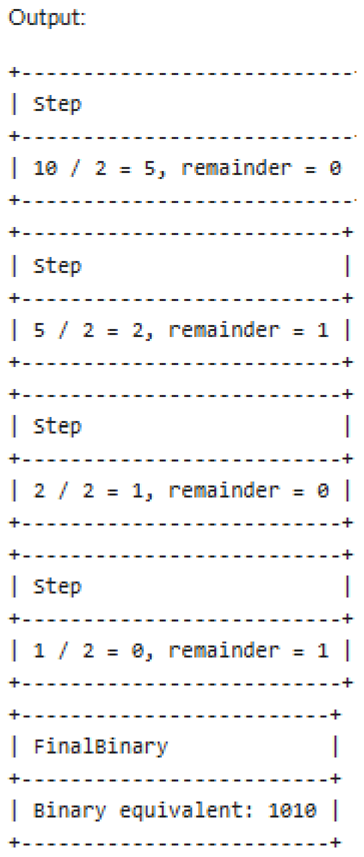
```
Binary equivalent of 10 is: 1010
```

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

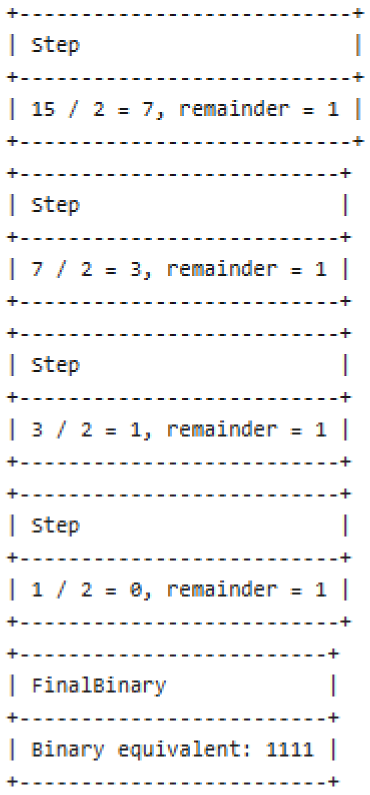
1. Modify the function to display step-by-step conversion while calculating binary.



Output:

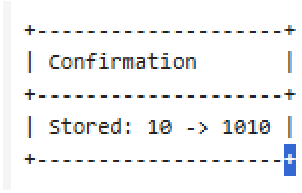
```
+---------------------------
| Step
+---------------------------
| 10 / 2 = 5, remainder = 0
+---------------------------
+--------------------------+
| Step                     |
+--------------------------+
| 5 / 2 = 2, remainder = 1 |
+--------------------------+
+--------------------------+
| Step                     |
+--------------------------+
| 2 / 2 = 1, remainder = 0 |
+--------------------------+
+--------------------------+
| Step                     |
+--------------------------+
| 1 / 2 = 0, remainder = 1 |
+--------------------------+
+------------------------+
| FinalBinary            |
+------------------------+
| Binary equivalent: 1010 |
+------------------------+
```

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



```
+---------------------------+
| Step                      |
+---------------------------+
| 15 / 2 = 7, remainder = 1 |
+---------------------------+
+--------------------------+
| Step                     |
+--------------------------+
| 7 / 2 = 3, remainder = 1 |
+--------------------------+
+--------------------------+
| Step                     |
+--------------------------+
| 3 / 2 = 1, remainder = 1 |
+--------------------------+
+--------------------------+
| Step                     |
+--------------------------+
| 1 / 2 = 0, remainder = 1 |
+--------------------------+
+-------------------------+
| FinalBinary             |
+-------------------------+
| Binary equivalent: 1111 |
+-------------------------+
```

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



```
+--------------------+
| Confirmation       |
+--------------------+
| Stored: 10 -> 1010 |
+--------------------+
```