PL/Sql-7 Create a stored function titled 'Age_calc'. Accept the date of birth of a person as a parameter. Calculate the age of the person in years, months and days e.g. 3 years, 2months, 10 days. Return the age in years directly (with the help of Return statement). The months and days are to be returned indirectly in the form of OUT parameters.

Step 1: Create Function to Return Years

```
DELIMITER //
CREATE FUNCTION Age_calc(dob DATE)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE v_years INT;
    SET v_years = TIMESTAMPDIFF(YEAR, dob, CURDATE());
    RETURN v_years;
END //
DELIMITER ;
```

Step 2: Create Procedure to Return Months and Days

```
DELIMITER //

CREATE PROCEDURE Get_Age_Full(
    IN p_dob DATE,
    OUT p_months INT,
    OUT p_days INT
)
BEGIN
    DECLARE v_years INT;
    DECLARE v_total_months INT;
    DECLARE v_remainder_days INT;
    DECLARE v_temp_date DATE;

    -- Step 1: Get years using function
    SET v_years = Age_calc(p_dob);

    -- Step 2: Calculate remaining months
    SET v_total_months = TIMESTAMPDIFF(MONTH, p_dob, CURDATE()) -
(v_years * 12);

    -- Step 3: Calculate remaining days
    SET v_temp_date = DATE_ADD(p_dob, INTERVAL v_years YEAR);
    SET v_temp_date = DATE_ADD(v_temp_date, INTERVAL v_total_months
MONTH);
    SET v_remainder_days = DATEDIFF(CURDATE(), v_temp_date);

    -- Step 4: Set OUT parameters
    SET p_months = v_total_months;
    SET p_days = v_remainder_days;
END //

DELIMITER ;
```

Step 3: Declare Variables to Hold OUT Parameters

```
SET @m = 0;
SET @d = 0;
```

Step 4: Call the Procedure with a DOB

```
CALL Get_Age_Full('2000-05-15', @m, @d);
```

Step 5: Get the Complete Age Output

```
SELECT Age_calc('2000-05-15') AS Years, @m AS Months, @d AS Days;
```

Expected Output (if today is 2025-11-05):

Years	Months	Days
25	5	21