Q1.

CREATE OR REPLACE FUNCTION CalculateAge(p\_DOB DATE)

RETURN NUMBER

IS

v\_Age NUMBER;

BEGIN

-- Calculate age based on the difference between the current date and the date of birth

v\_Age := TRUNC(MONTHS\_BETWEEN(SYSDATE, p\_DOB) / 12);

RETURN v\_Age;

END CalculateAge;

/

Q2.

CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(

p\_LoanAmount NUMBER,

p\_InterestRate NUMBER, -- Annual interest rate as a percentage

p\_DurationYears NUMBER -- Duration of the loan in years

)

RETURN NUMBER

IS

v\_MonthlyRate NUMBER; -- Monthly interest rate

v\_NumberOfMonths NUMBER; -- Total number of payments (months)

v\_EMI NUMBER; -- Equated Monthly Installment (EMI)

BEGIN

-- Calculate the monthly interest rate (as a fraction)

v\_MonthlyRate := p\_InterestRate / 1200;

-- Calculate the total number of months

v\_NumberOfMonths := p\_DurationYears \* 12;

-- Calculate the EMI using the formula

IF v\_MonthlyRate != 0 THEN

v\_EMI := p\_LoanAmount \* v\_MonthlyRate \* POWER(1 + v\_MonthlyRate, v\_NumberOfMonths) /

(POWER(1 + v\_MonthlyRate, v\_NumberOfMonths) - 1);

ELSE

-- If the interest rate is 0, the EMI is simply the principal divided by the number of months

v\_EMI := p\_LoanAmount / v\_NumberOfMonths;

END IF;

RETURN v\_EMI;

END CalculateMonthlyInstallment;

/

Q3.

CREATE OR REPLACE FUNCTION HasSufficientBalance(

p\_AccountID NUMBER,

p\_Amount NUMBER

)

RETURN NUMBER

IS

v\_Balance NUMBER;

BEGIN

-- Retrieve the balance of the specified account

SELECT Balance

INTO v\_Balance

FROM Accounts

WHERE AccountID = p\_AccountID;

-- Check if the balance is greater than or equal to the specified amount

IF v\_Balance >= p\_Amount THEN

RETURN 1; -- Sufficient balance (TRUE)

ELSE

RETURN 0; -- Insufficient balance (FALSE)

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- If the account does not exist, return 0 (FALSE)

RETURN 0;

WHEN OTHERS THEN

-- Handle other potential exceptions

RETURN 0;

END HasSufficientBalance;

/