**Exercise 4: Functions**

**Scenario 1: Calculate the age of customers for eligibility checks.**

**Question:** Write a function CalculateAge that takes a customer's date of birth as input and returns their age in years.

**Function:**

SQL> CREATE OR REPLACE FUNCTION CalculateAge(p\_dob DATE)

2 RETURN NUMBER

3 IS

4 v\_age NUMBER;

5 BEGIN

6 v\_age := FLOOR(MONTHS\_BETWEEN(SYSDATE, p\_dob) / 12);

7 RETURN v\_age;

8 END;

9 /

**Scenario 2: The bank needs to compute the monthly installment for a loan.**

**Question:** Write a function **CalculateMonthlyInstallment** that takes the loan amount, interest rate, and loan duration in years as input and returns the monthly installment amount.

**Function:**

SQL> CREATE OR REPLACE FUNCTION CalculateMonthlyInstallment(p\_loan\_amount NUMBER, p\_interest\_rate NUMBER, p\_duration\_years NUMBER)

2 RETURN NUMBER

3 IS

4 v\_monthly\_installment NUMBER;

5 v\_monthly\_interest\_rate NUMBER;

6 v\_total\_months NUMBER;

7 BEGIN

8 v\_monthly\_interest\_rate := p\_interest\_rate / 12 / 100;

9 v\_total\_months := p\_duration\_years \* 12;

10

11 IF v\_monthly\_interest\_rate = 0 THEN

12 v\_monthly\_installment := p\_loan\_amount / v\_total\_months;

13 ELSE

14 v\_monthly\_installment := p\_loan\_amount \* (v\_monthly\_interest\_rate \* POWER(1 + v\_monthly\_interest\_rate, v\_total\_months)) / (POWER(1 + v\_monthly\_interest\_rate, v\_total\_months) - 1);

15 END IF;

16

17 RETURN v\_monthly\_installment;

18 END;

19 /

**Scenario 3: Check if a customer has sufficient balance before making a transaction.**

**Question:** Write a function HasSufficientBalance that takes an account ID and an amount as input and returns a boolean indicating whether the account has at least the specified amount.

**Function:**

SQL> CREATE OR REPLACE FUNCTION HasSufficientBalance(p\_account\_id NUMBER, p\_amount NUMBER)

2 RETURN NUMBER

3 IS

4 v\_balance NUMBER;

5 BEGIN

6 SELECT Balance INTO v\_balance

7 FROM Accounts

8 WHERE AccountID = p\_account\_id;

9

10 RETURN CASE

11 WHEN v\_balance >= p\_amount THEN 1

12 ELSE 0

13 END;

14 EXCEPTION

15 WHEN NO\_DATA\_FOUND THEN

16 RETURN 0;

17 END;

18 /