PL/SQL Exercise Solutions - Exercises 1 & 3

Exercise 1: Control Structures

```
Exercise 1: Control Structures

Scenario 1: Apply 1% discount to loan interest rates for customers above 60
BEGIN
    FOR rec IN (SELECT LoanID, CustomerID, InterestRate, DOB
                FROM Loans
                JOIN Customers USING (CustomerID))
    LOOP
        IF MONTHS_BETWEEN(SYSDATE, rec.DOB) / 12 > 60 THEN
            UPDATE Loans
            SET InterestRate = InterestRate - 1
            WHERE LoanID = rec.LoanID;
        END IF;
    END LOOP;
END;
/

Scenario 2: Set IsVIP flag for balances over $10,000
ALTER TABLE Customers ADD IsVIP CHAR(1);

BEGIN
    FOR rec IN (SELECT CustomerID, Balance FROM Customers) LOOP
        IF rec.Balance > 10000 THEN
            UPDATE Customers
            SET IsVIP = 'Y'
            WHERE CustomerID = rec.CustomerID;
        END IF;
    END LOOP;
END;
/

Scenario 3: Reminders for loans due in next 30 days
BEGIN
    FOR rec IN (
        SELECT c.Name, l.LoanID, l.EndDate
        FROM Loans l
        JOIN Customers c ON l.CustomerID = c.CustomerID
        WHERE l.EndDate BETWEEN SYSDATE AND SYSDATE + 30
    )
    LOOP
         DBMS_OUTPUT.PUT_LINE('Reminder: Loan ID ' || rec.LoanID || ' for ' || rec.Name
||
                              ' is due on ' || TO_CHAR(rec.EndDate, 'YYYY-MM-DD'));
    END LOOP;
END;
/
```

Exercise 3: Stored Procedures

```
Exercise 3: Stored Procedures

Scenario 1: ProcessMonthlyInterest
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
BEGIN
    UPDATE Accounts
    SET Balance = Balance + (Balance * 0.01)
    WHERE AccountType = 'Savings';
END;
/

Scenario 2: UpdateEmployeeBonus
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(
    p_department IN VARCHAR2,
    p_bonus_pct IN NUMBER
) IS
BEGIN
    UPDATE Employees
    SET Salary = Salary + (Salary * p_bonus_pct / 100)
    WHERE Department = p_department;
END;
/

Scenario 3: TransferFunds
CREATE OR REPLACE PROCEDURE TransferFunds(
    p_from_account IN NUMBER,
    p_to_account IN NUMBER,
    p_amount IN NUMBER
) IS
```

PL/SQL Exercise Solutions - Exercises 1 & 3

```
    v_balance NUMBER;
BEGIN
    SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = p_from_account;

    IF v_balance < p_amount THEN
        RAISE_APPLICATION_ERROR(-20001, 'Insufficient funds in source account.');
    END IF;

    UPDATE Accounts
    SET Balance = Balance - p_amount
    WHERE AccountID = p_from_account;

    UPDATE Accounts
    SET Balance = Balance + p_amount
    WHERE AccountID = p_to_account;

    COMMIT;
END;
/
```