Mandatory Hands-on Exercises PL/SQL programming

Exercise 1:

Scenario 1:

```
BEGIN
  FOR c IN (SELECT CustomerID, DOB FROM Customers) LOOP
    IF MONTHS_BETWEEN(SYSDATE, c.DOB) / 12 > 60 THEN
    UPDATE Loans
    SET InterestRate = InterestRate – 1
    WHERE CustomerID = c.CustomerID;
    END IF;
  END LOOP;
  COMMIT;
END;
```

Scenario 2:

```
ALTER TABLE Customers ADD IsVIP CHAR(1);
BEGIN
  FOR c IN (SELECT CustomerID, Balance FROM Customers) LOOP
    IF c.Balance > 10000 THEN
    UPDATE Customers
    SET IsVIP = 'Y'
    WHERE CustomerID = c.CustomerID;
    ELSE
    UPDATE Customers
    SET IsVIP = 'N'
    WHERE CustomerID = c.CustomerID
    END IF;
  END LOOP;
  COMMIT;
```

```
END;
```

Scenario 3:

```
DECLARE
   v_name Customers.Name%TYPE;
BEGIN
   FOR l IN (
      SELECT LoanID, CustomerID, EndDate
      FROM Loans
      WHERE EndDate BETWEEN SYSDATE AND SYSDATE + 30
   ) LOOP
      SELECT Name INTO v_name FROM Customers WHERE CustomerID = l.CustomerID;
      DBMS_OUTPUT.PUT_LINE('Reminder: Loan ID ' || l.LoanID || ' for customer ' || v_name || ' is due on ' || TO_CHAR(l.EndDate, 'DD-MON-YYYY'));
   END LOOP;
END;
```

Exercise 3:

Scenario 1:

```
CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS
BEGIN
   FOR acc IN (SELECT AccountID, Balance FROM Accounts WHERE AccountType = 'Savings') LOOP
      UPDATE Accounts
      SET Balance = acc.Balance + (acc.Balance * 0.01)
      WHERE AccountID = acc.AccountID;
   END LOOP;
   COMMIT;
END;
```

Scenario 2:

```
CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (
   dept_name IN VARCHAR2,
   bonus_percent IN NUMBER
) IS
BEGIN
   FOR emp IN (SELECT EmployeeID, Salary FROM Employees WHERE Department = dept_name) LOOP
      UPDATE Employees
      SET Salary = emp.Salary + (emp.Salary * bonus_percent / 100)
      WHERE EmployeeID = emp.EmployeeID;
   END LOOP;
   COMMIT;
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Error occurred: ' || SQLERRM);
END;
```

Scenario 3:

```
CREATE OR REPLACE PROCEDURE TransferFunds (
   from_account IN NUMBER,
   to_account IN NUMBER,
   amount IN NUMBER
) IS
   source_balance NUMBER;
BEGIN
   SELECT Balance INTO source_balance FROM Accounts WHERE AccountID = from_account FOR UPDATE;

   IF source_balance >= amount THEN
```

```
    UPDATE Accounts
    SET Balance = Balance - amount
    WHERE AccountID = from_account;

    UPDATE Accounts
    SET Balance = Balance + amount
    WHERE AccountID = to_account;

    COMMIT;
  ELSE
    RAISE_APPLICATION_ERROR(-20001, 'Insufficient balance in source account');
  END IF;
END;
```