Exercise1:

BEGIN

FOR rec IN (

SELECT c.CustomerID, l.LoanID, l.InterestRate

FROM Customers c

JOIN Loans l ON c.CustomerID = l.CustomerID

WHERE c.Age > 60

)

LOOP

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE LoanID = rec.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Applied 1% discount to Loan ID ' || rec.LoanID || ' for Customer ID ' || rec.CustomerID);

END LOOP;

END;

Output--Applied 1% discount to Loan ID 101 for Customer ID 1

Applied 1% discount to Loan ID 104 for Customer ID 3

Scenario2: BEGIN

FOR rec IN (

SELECT CustomerID, Balance

FROM Customers

WHERE Balance > 10000 AND IsVIP = 'FALSE'

)

LOOP

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = rec.CustomerID;

DBMS\_OUTPUT.PUT\_LINE('Customer ID ' || rec.CustomerID || ' promoted to VIP status.');

END LOOP;

END;

OUTPUT-Customer ID 2 promoted to VIP status.

Customer ID 5 promoted to VIP status.

Scenario3:: BEGIN

FOR rec IN (

SELECT l.LoanID, c.Name, l.DueDate

FROM Loans l

JOIN Customers c ON c.CustomerID = l.CustomerID

WHERE l.DueDate BETWEEN SYSDATE AND SYSDATE + 30

)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.LoanID || ' for customer ' || rec.Name ||

' is due on ' || TO\_CHAR(rec.DueDate, 'DD-MON-YYYY'));

END LOOP;

END;

OUTPUT-

Reminder: Loan ID 107 for customer Alice is due on 15-JUL-2025

Reminder: Loan ID 112 for customer Bob is due on 05-JUL-2025

EXERCISE 3::

Scenario1::

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

FOR rec IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'SAVINGS'

)

LOOP

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountID = rec.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Interest applied to Account ID ' || rec.AccountID);

END LOOP;

END;

OUTPUT-Interest applied to Account ID 1001

Interest applied to Account ID 1003

Scenario2:: CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_dept\_id IN NUMBER,

p\_bonus\_pct IN NUMBER -- Example: pass 10 for 10%

) AS

BEGIN

FOR rec IN (

SELECT EmployeeID, Salary

FROM Employees

WHERE DepartmentID = p\_dept\_id

)

LOOP

UPDATE Employees

SET Salary = Salary + (Salary \* (p\_bonus\_pct / 100))

WHERE EmployeeID = rec.EmployeeID;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to Employee ID ' || rec.EmployeeID);

END LOOP;

END;

Output::

Bonus applied to Employee ID 2001

Bonus applied to Employee ID 2002

Scenario 3:: CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_id IN NUMBER,

p\_to\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Get current balance of source account

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = p\_from\_id;

-- Check sufficient funds

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in source account.');

END IF;

-- Perform transfer

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_from\_id;

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_to\_id;

DBMS\_OUTPUT.PUT\_LINE('Transferred $' || p\_amount || ' from Account ' || p\_from\_id || ' to Account ' || p\_to\_id);

END;

OUTPUT-Transferred $500 from Account 1001 to Account 1002