**PL/SQL:-**

**Exercise - 1:**

BEGIN

FOR rec IN (

SELECT c.customer\_id, l.loan\_id

FROM customers c

JOIN loans l ON c.customer\_id = l.customer\_id

WHERE c.age > 60

) LOOP

UPDATE loans

SET interest\_rate = interest\_rate - 1

WHERE loan\_id = rec.loan\_id;

END LOOP;

FOR rec IN (SELECT customer\_id FROM customers WHERE balance > 10000) LOOP

UPDATE customers

SET isvip = 'Y'

WHERE customer\_id = rec.customer\_id;

END LOOP;

FOR rec IN (

SELECT l.loan\_id, c.name, l.due\_date

FROM loans l

JOIN customers c ON l.customer\_id = c.customer\_id

WHERE l.due\_date BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || rec.loan\_id ||

' for customer ' || rec.name ||

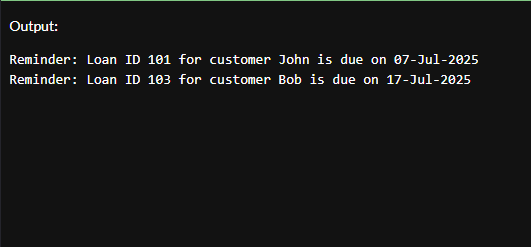
' is due on ' || TO\_CHAR(rec.due\_date, 'DD-Mon-YYYY'));

END LOOP;

COMMIT;

END;

/

****

**Exercise - 3**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE ACCOUNTS

SET Balance = Balance + (Balance \* 0.01)

WHERE UPPER(AccountType) = 'SAVINGS';

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' savings accounts updated with interest.');

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_Department IN VARCHAR2,

p\_BonusPercent IN NUMBER

) IS

BEGIN

UPDATE EMPLOYEES

SET Salary = Salary + (Salary \* p\_BonusPercent / 100)

WHERE UPPER(Department) = UPPER(p\_Department);

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' employees in ' || p\_Department || ' got a ' || p\_BonusPercent || '% bonus.');

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) IS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance

FROM ACCOUNTS

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

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

END IF;

UPDATE ACCOUNTS

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

UPDATE ACCOUNTS

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_Amount || ' from account ' || p\_FromAccountID || ' to account ' || p\_ToAccountID);

COMMIT;

END;

/

