**Exercise 3: Stored Procedures**

**Ans 1.**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

UPDATE Accounts

SET Balance = (Balance \* 1.01)

WHERE AccountType = 'Savings';

COMMIT;

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

SELECT \* FROM Accounts;

**Ans 2.**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

deptName IN Employees.Department%TYPE,

bonusPercent IN NUMBER

) IS

BEGIN

UPDATE Employees

SET Salary = Salary \* ((100 + bonusPercent)/100)

WHERE Department = deptName;

COMMIT;

END;

/

DECLARE

p\_deptName VARCHAR2(50) := '&deptName';

p\_bonusPercent NUMBER := &bonusPercent;

BEGIN

UpdateEmployeeBonus(p\_deptName, p\_bonusPercent);

END;

/

**Ans 3.**

CREATE OR REPLACE PROCEDURE TransferFunds (

fromId IN Accounts.AccountID%TYPE,

toId IN Accounts.AccountID%TYPE,

amount IN NUMBER

) IS

v\_fromBal Accounts.Balance%TYPE;

v\_toBal Accounts.Balance%TYPE;

insufficientFunds EXCEPTION;

BEGIN

SELECT BALANCE INTO v\_fromBal

FROM Accounts

WHERE AccountID = fromId

FOR UPDATE;

SELECT BALANCE INTO v\_toBal

FROM Accounts

WHERE AccountID = toId

FOR UPDATE;

IF amount > v\_fromBal THEN

RAISE insufficientFunds;

ELSE

UPDATE Accounts

SET BALANCE = v\_fromBal - amount

WHERE AccountID = fromId;

UPDATE Accounts

SET BALANCE = v\_toBal + amount

WHERE AccountID = toId;

COMMIT;

END IF;

EXCEPTION

WHEN insufficientFunds THEN

DBMS\_OUTPUT.PUT\_LINE('INSUFFICIENT BALANCE!');

ROLLBACK;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('AN UNEXPECTED ERROR OCCURED:' || SQLERRM);

ROLLBACK;

END;

/

DECLARE

p\_fromId NUMBER:=&fromId;

p\_toId NUMBER:=&toId;

p\_amount NUMBER:=&amount;

BEGIN

TransferFunds(p\_fromId, p\_toId, p\_amount);

END;

/