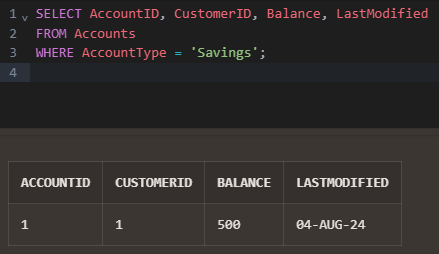
Q1.

*Initial Accounts Table:*

**

*PL/SQL Block:*

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

-- Update the balance for all savings accounts

UPDATE Accounts

SET Balance = Balance \* 1.01,

LastModified = SYSDATE

WHERE AccountType = 'Savings';

-- Commit the transaction

COMMIT;

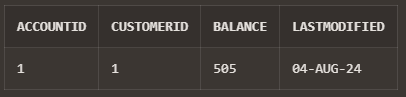
-- Output a message indicating the process is complete

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to all savings accounts.');

END ProcessMonthlyInterest;

/

*Output:*

**

Q2.

*Initial Employees Table:*

**

*PL/SQL Block:*

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_Department IN VARCHAR2,

p\_BonusPercentage IN NUMBER

)

IS

BEGIN

-- Update the salary for all employees in the given department

UPDATE Employees

SET Salary = Salary \* (1 + p\_BonusPercentage / 100)

WHERE Department = p\_Department;

-- Commit the transaction

COMMIT;

-- Output a message indicating the process is complete

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to all employees in department ' || p\_Department || '.');

END UpdateEmployeeBonus;

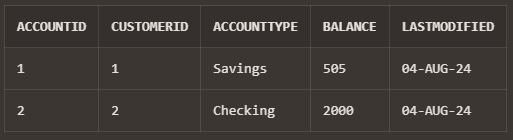
/

*Output:*



Q3.

*Initial Accounts Table:*



*PL/SQL Block:*

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_SourceAccountID IN NUMBER,

p\_DestinationAccountID IN NUMBER,

p\_Amount IN NUMBER

)

IS

v\_SourceBalance NUMBER;

v\_MaxTransactionID NUMBER;

BEGIN

-- Check if the source account has sufficient balance

SELECT Balance INTO v\_SourceBalance

FROM Accounts

WHERE AccountID = p\_SourceAccountID

FOR UPDATE;

IF v\_SourceBalance < p\_Amount THEN

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

END IF;

-- Deduct the amount from the source account

UPDATE Accounts

SET Balance = Balance - p\_Amount, LastModified = SYSDATE

WHERE AccountID = p\_SourceAccountID;

-- Add the amount to the destination account

UPDATE Accounts

SET Balance = Balance + p\_Amount, LastModified = SYSDATE

WHERE AccountID = p\_DestinationAccountID;

-- Get the max TransactionID to generate new IDs

SELECT NVL(MAX(TransactionID), 0) INTO v\_MaxTransactionID

FROM Transactions;

-- Record the transaction for the source account (Withdrawal)

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (v\_MaxTransactionID + 1, p\_SourceAccountID, SYSDATE, p\_Amount, 'Withdrawal');

-- Record the transaction for the destination account (Deposit)

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (v\_MaxTransactionID + 2, p\_DestinationAccountID, SYSDATE, p\_Amount, 'Deposit');

-- Commit the transaction

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Rollback the transaction in case of error

ROLLBACK;

RAISE;

END TransferFunds;

/

*Output:*

