**Exercise 3: Stored Procedures**

INSERT INTO Accounts VALUES (3, 102, 'Savings', 15000, SYSDATE);

**Scenario 1:**

**Question:** Write a stored procedure **ProcessMonthlyInterest** that calculates and updates the balance of all savings accounts by applying an interest rate of 1% to the current balance.

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

  UPDATE Accounts

  SET Balance = Balance + (Balance \* 0.01)

  WHERE AccountType = 'Savings';

  COMMIT;

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

EXCEPTION

  WHEN OTHERS THEN

    ROLLBACK;

    DBMS\_OUTPUT.PUT\_LINE('Interest application failed: ' || SQLERRM);

END;

/

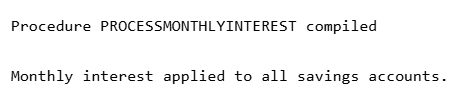
BEGIN

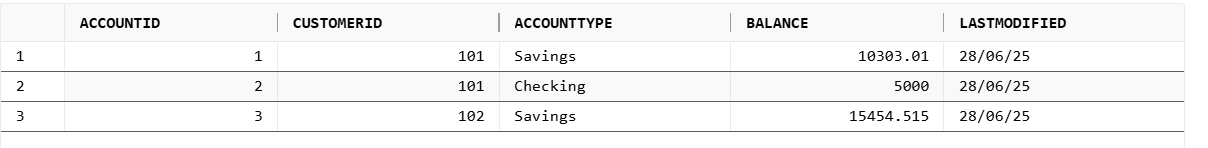
  ProcessMonthlyInterest;

END;

/

**Output**





**Scenario 2:**

**Question:** Write a stored procedure **UpdateEmployeeBonus** that updates the salary of employees in a given department by adding a bonus percentage passed as a parameter.

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

  p\_department IN VARCHAR2,

  p\_bonus\_percent IN NUMBER

)

IS

BEGIN

  UPDATE Employees

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

  WHERE Department = p\_department;

  IF SQL%ROWCOUNT = 0 THEN

    RAISE\_APPLICATION\_ERROR(-20002, 'No employees found in department: ' || p\_department);

  END IF;

  COMMIT;

  DBMS\_OUTPUT.PUT\_LINE('✅ Bonus applied to department: ' || p\_department);

EXCEPTION

  WHEN OTHERS THEN

    ROLLBACK;

    DBMS\_OUTPUT.PUT\_LINE('❌ Bonus update failed: ' || SQLERRM);

END;

/

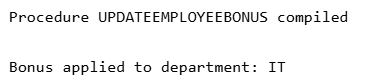
BEGIN

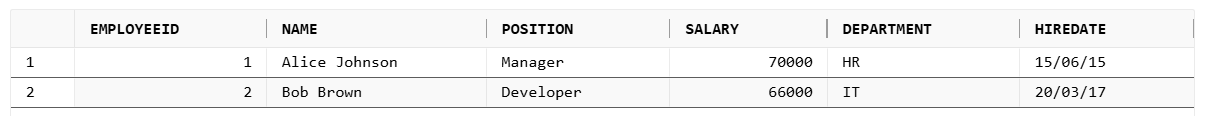
  TransferFunds(1, 2, 3000);

END;

/

**Output**





**Scenario 3:**

**Question:** Write a stored procedure **TransferFunds** that transfers a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CREATE OR REPLACE PROCEDURE TransferFunds (

  p\_from\_account IN NUMBER,

  p\_to\_account   IN NUMBER,

  p\_amount       IN NUMBER

)

IS

  v\_balance NUMBER;

BEGIN

  SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from\_account FOR UPDATE;

IF v\_balance < p\_amount THEN

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

  END IF;

  UPDATE Accounts SET Balance = Balance - p\_amount WHERE AccountID = p\_from\_account;

  UPDATE Accounts SET Balance = Balance + p\_amount WHERE AccountID = p\_to\_account;

COMMIT;

  DBMS\_OUTPUT.PUT\_LINE('✅ Funds transferred successfully.');

EXCEPTION

  WHEN OTHERS THEN

    ROLLBACK;

    DBMS\_OUTPUT.PUT\_LINE('❌ Fund transfer failed: ' || SQLERRM);

END;

/

BEGIN

  TransferFunds(1, 2, 3000);

END;

/

**Output**

