**Exercise 3: Stored Procedures**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

  FOR rec IN (

    SELECT AccountID, Balance

    FROM Accounts

    WHERE AccountType = 'Savings'

  ) LOOP

    UPDATE Accounts

    SET Balance = rec.Balance + (rec.Balance \* 0.01)

    WHERE AccountID = rec.AccountID;

  END LOOP;

  COMMIT;

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

END;

/

BEGIN

  ProcessMonthlyInterest;

END;

/

SELECT \* FROM Accounts;

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

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

  ELSE

    DBMS\_OUTPUT.PUT\_LINE('No employees found in ' || p\_department || ' department.');

  END IF;

  COMMIT;

END;

/

BEGIN

  UpdateEmployeeBonus('IT', 10);

END;

/

SELECT \* FROM Employees;

CREATE OR REPLACE PROCEDURE TransferFunds (

    p\_from\_account\_id IN NUMBER,

    p\_to\_account\_id IN NUMBER,

    p\_amount IN NUMBER

)

IS

    v\_balance NUMBER;

BEGIN

  SELECT Balance INTO v\_balance

  FROM Accounts

  WHERE AccountID = p\_from\_account\_id;

  IF v\_balance < p\_amount THEN

    RAISE\_APPLICATION\_ERROR(-20003, 'Insufficient balance in source account.');

  END IF;

  UPDATE Accounts

  SET Balance = Balance - p\_amount

  WHERE AccountID = p\_from\_account\_id;

  UPDATE Accounts

  SET Balance = Balance + p\_amount

  WHERE AccountID = p\_to\_account\_id;

  COMMIT;

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

EXCEPTION

  WHEN OTHERS THEN

    ROLLBACK;

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

END;

/

BEGIN

  TransferFunds(1, 2, 300);

END;

/

SELECT \* FROM Accounts;

















