PL/SQL Stored Procedures Exercise

# Setup Tables and Sample Data

BEGIN  
EXECUTE IMMEDIATE 'DROP TABLE accounts';  
EXECUTE IMMEDIATE 'DROP TABLE employees';  
EXCEPTION WHEN OTHERS THEN NULL;  
END;  
/  
  
CREATE TABLE accounts (  
account\_id NUMBER PRIMARY KEY,  
customer\_name VARCHAR2(50),  
balance NUMBER,  
account\_type VARCHAR2(20)  
);  
  
CREATE TABLE employees (  
employee\_id NUMBER PRIMARY KEY,  
name VARCHAR2(50),  
department VARCHAR2(30),  
salary NUMBER  
);  
  
BEGIN  
INSERT INTO accounts VALUES (1, 'Adnan', 5000, 'savings');  
INSERT INTO accounts VALUES (2, 'Siva', 8000, 'current');  
INSERT INTO accounts VALUES (3, 'Somu', 10000, 'savings');  
INSERT INTO accounts VALUES (4, 'Rafi', 3000, 'savings');  
INSERT INTO accounts VALUES (5, 'Adnan', 2000, 'current');  
COMMIT;  
END;  
/  
  
BEGIN  
INSERT INTO employees VALUES (101, 'Adnan', 'IT', 50000);  
INSERT INTO employees VALUES (102, 'Siva', 'HR', 45000);  
INSERT INTO employees VALUES (103, 'Somu', 'IT', 55000);  
INSERT INTO employees VALUES (104, 'Rafi', 'Finance', 40000);  
COMMIT;  
END;/

# Scenario 1: ProcessMonthlyInterest Procedure

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS  
BEGIN  
 FOR acc IN (SELECT account\_id, balance FROM accounts WHERE account\_type = 'savings') LOOP  
 UPDATE accounts SET balance = balance + (balance \* 0.01) WHERE account\_id = acc.account\_id;  
 END LOOP;  
 DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to savings accounts.');  
END;  
/  
  
BEGIN  
 ProcessMonthlyInterest;  
END;  
/

# Scenario 2: UpdateEmployeeBonus Procedure

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_dept VARCHAR2, p\_bonus\_pct NUMBER) IS  
BEGIN  
 FOR emp IN (SELECT employee\_id FROM employees WHERE department = p\_dept) LOOP  
 UPDATE employees SET salary = salary + (salary \* p\_bonus\_pct / 100) WHERE employee\_id = emp.employee\_id;  
 END LOOP;  
 DBMS\_OUTPUT.PUT\_LINE('Bonus updated for department: ' || p\_dept);  
END;  
/  
  
BEGIN  
 UpdateEmployeeBonus('IT', 10);  
END;  
/

# Scenario 3: TransferFunds Procedure

CREATE OR REPLACE PROCEDURE TransferFunds(p\_from NUMBER, p\_to NUMBER, p\_amount NUMBER) IS  
v\_balance NUMBER;  
BEGIN  
 SELECT balance INTO v\_balance FROM accounts WHERE account\_id = p\_from FOR UPDATE;  
 IF v\_balance < p\_amount THEN  
 DBMS\_OUTPUT.PUT\_LINE('Insufficient balance in source account.');  
 ELSE  
 UPDATE accounts SET balance = balance - p\_amount WHERE account\_id = p\_from;  
 UPDATE accounts SET balance = balance + p\_amount WHERE account\_id = p\_to;  
 DBMS\_OUTPUT.PUT\_LINE('Transfer successful.');  
 END IF;  
END;  
/  
  
BEGIN  
 TransferFunds(1, 2, 1000);  
END;  
/

# Final Output

SELECT \* FROM accounts;  
SELECT \* FROM employees;