**Exercise 3: Stored Procedures**

**CODE**

**Create tables**

-- Table for accounts

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

account\_type VARCHAR2(20),

balance NUMBER

);

-- Table for employees

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

department\_id NUMBER,

salary NUMBER

);

/

**Insert Sample Data**

BEGIN

-- Insert into accounts

INSERT INTO accounts VALUES (101, 'SAVINGS', 1000);

INSERT INTO accounts VALUES (102, 'SAVINGS', 2000);

INSERT INTO accounts VALUES (103, 'CURRENT', 3000);

INSERT INTO accounts VALUES (104, 'SAVINGS', 4000);

-- Insert into employees

INSERT INTO employees VALUES (201, 10, 5000);

INSERT INTO employees VALUES (202, 10, 6000);

INSERT INTO employees VALUES (203, 20, 7000);

COMMIT;

END;

/

**Scenario 1 ─ Stored procedure to process monthly interest**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_type = 'SAVINGS';

COMMIT;

END;

/

**Scenario 2 ─ Stored procedure to apply bonus to a department**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN NUMBER,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

UPDATE employees

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

WHERE department\_id = p\_department\_id;

COMMIT;

END;

/

**Scenario 3 ─ Stored procedure for fund transfer**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_account

FOR UPDATE;

IF v\_balance >= p\_amount THEN

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account;

COMMIT;

ELSE

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

END IF;

END;

/

**ProcessMonthlyInterest**

BEGIN

ProcessMonthlyInterest;

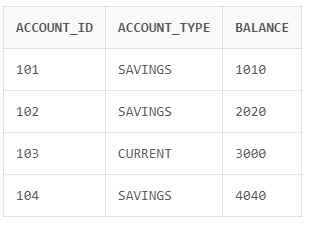
END;

/

-- View updated balances

SELECT \* FROM accounts;

**OUTPUT**

****

**Test 2: UpdateEmployeeBonus**

Add 10% bonus to department 10

BEGIN

UpdateEmployeeBonus(10, 10);

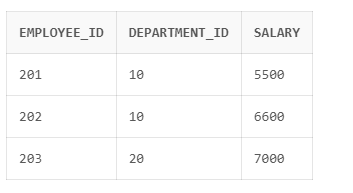
END;

/

-- View updated salaries

SELECT \* FROM employees;

**OUTPUT**



**Test 3: TransferFunds**

Transfer 500 from account 104 to account 103:

BEGIN

TransferFunds(104, 103, 500);

END;

/

-- View updated balances

SELECT \* FROM accounts;

**OUTPUT**

