**Exercise 3: Stored Procedures**

**Scenario 1:** The bank needs to process monthly interest for all savings accounts.

**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.

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER,

account\_type VARCHAR2(20)

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

salary NUMBER,

department\_id NUMBER

);

INSERT INTO accounts VALUES (101, 1, 10000, 'SAVINGS');

INSERT INTO accounts VALUES (102, 2, 20000, 'SAVINGS');

INSERT INTO accounts VALUES (103, 3, 15000, 'CURRENT');

INSERT INTO employees VALUES (1, 'Sanju', 30000, 10);

INSERT INTO employees VALUES (2, 'Mohan', 35000, 10);

INSERT INTO employees VALUES (3, 'Suga', 40000, 20);

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

v\_count NUMBER;

BEGIN

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_type = 'SAVINGS';

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

v\_count := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to ' || v\_count || ' savings account(s).');

COMMIT;

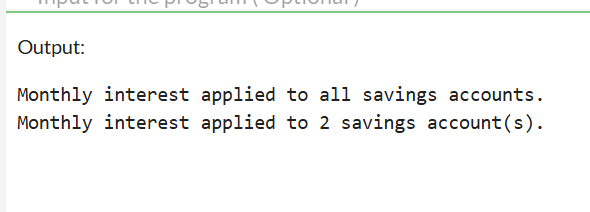
END;

/

SET SERVEROUTPUT ON;

EXEC ProcessMonthlyInterest;

**OUTPUT:**

****

**Scenario 2:** The bank wants to implement a bonus scheme for employees based on their performance.

**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 TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER,

account\_type VARCHAR2(20)

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

salary NUMBER,

department\_id NUMBER

);

INSERT INTO accounts VALUES (101, 1, 10000, 'SAVINGS');

INSERT INTO accounts VALUES (102, 2, 20000, 'SAVINGS');

INSERT INTO accounts VALUES (103, 3, 15000, 'CURRENT');

INSERT INTO employees VALUES (1, 'Sanju', 30000, 10);

INSERT INTO employees VALUES (2, 'Mohan', 35000, 10);

INSERT INTO employees VALUES (3, 'Suga', 40000, 20);

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_id IN NUMBER,

bonus\_percent IN NUMBER

) AS

v\_updated\_count NUMBER;

BEGIN

UPDATE employees

SET salary = salary + (salary \* bonus\_percent / 100)

WHERE department\_id = dept\_id;

v\_updated\_count := SQL%ROWCOUNT;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || bonus\_percent || '% applied to ' ||

v\_updated\_count || ' employee(s) in department ' || dept\_id);

COMMIT;

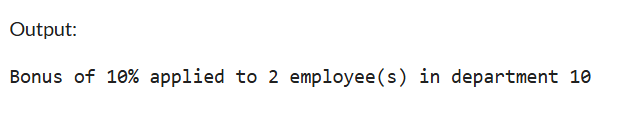
END;

/

SET SERVEROUTPUT ON;

EXEC UpdateEmployeeBonus(10, 10);

**OUTPUT:**

****

**Scenario 3:** Customers should be able to transfer funds between their accounts.

**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 TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER,

account\_type VARCHAR2(20)

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

name VARCHAR2(100),

salary NUMBER,

department\_id NUMBER

);

INSERT INTO accounts VALUES (101, 1, 10000, 'SAVINGS');

INSERT INTO accounts VALUES (102, 2, 20000, 'SAVINGS');

INSERT INTO accounts VALUES (103, 3, 15000, 'CURRENT');

INSERT INTO employees VALUES (1, 'Sanju', 30000, 10);

INSERT INTO employees VALUES (2, 'Mohan', 35000, 10);

INSERT INTO employees VALUES (3, 'Suga', 40000, 20);

SET SERVEROUTPUT ON;

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_acc\_id IN NUMBER,

to\_acc\_id IN NUMBER,

amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = from\_acc\_id;

IF v\_balance >= amount THEN

UPDATE accounts

SET balance = balance - amount

WHERE account\_id = from\_acc\_id;

UPDATE accounts

SET balance = balance + amount

WHERE account\_id = to\_acc\_id;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer of Rs' || amount ||

' from Account ' || from\_acc\_id ||

' to Account ' || to\_acc\_id ||

' was successful.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance. Transfer failed.');

END IF;

END;

/

SET SERVEROUTPUT ON;

EXEC TransferFunds(101, 102, 500);

