**PL SQL**

**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.

**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.

**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.

**CODE:**

CREATE TABLE savings\_accounts (

account\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(50),

balance NUMBER

);

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

department VARCHAR2(30),

salary NUMBER

);

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(50),

balance NUMBER

);

BEGIN

INSERT INTO savings\_accounts VALUES (101, 'Hari', 5000);

INSERT INTO savings\_accounts VALUES (102, 'Rahul', 3000);

INSERT INTO employees VALUES (1, 'Asha', 'HR', 40000);

INSERT INTO employees VALUES (2, 'Vikram', 'IT', 50000);

INSERT INTO employees VALUES (3, 'Meera', 'HR', 45000);

INSERT INTO accounts VALUES (201, 'Hari', 8000);

INSERT INTO accounts VALUES (202, 'Hari', 2000);

COMMIT;

END;

/

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE savings\_accounts

SET balance = balance + (balance \* 0.01);

END;

/

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_dept IN VARCHAR2, p\_bonus\_percent IN NUMBER) IS

BEGIN

UPDATE employees

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

WHERE department = p\_dept;

END;

/

CREATE OR REPLACE PROCEDURE TransferFunds(p\_from\_acct IN NUMBER, p\_to\_acct IN NUMBER, p\_amount IN NUMBER) IS

v\_balance NUMBER;

BEGIN

SELECT balance INTO v\_balance FROM accounts WHERE account\_id = p\_from\_acct FOR UPDATE;

IF v\_balance < p\_amount THEN

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

RETURN;

END IF;

UPDATE accounts SET balance = balance - p\_amount WHERE account\_id = p\_from\_acct;

UPDATE accounts SET balance = balance + p\_amount WHERE account\_id = p\_to\_acct;

END;

/

BEGIN

ProcessMonthlyInterest;

UpdateEmployeeBonus('HR', 10);

TransferFunds(201, 202, 1500);

DBMS\_OUTPUT.PUT\_LINE('Savings Accounts:');

FOR r IN (SELECT \* FROM savings\_accounts) LOOP

DBMS\_OUTPUT.PUT\_LINE(r.account\_id || ' - ' || r.customer\_name || ' - Balance: ' || r.balance);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'Employees:');

FOR r IN (SELECT \* FROM employees) LOOP

DBMS\_OUTPUT.PUT\_LINE(r.emp\_id || ' - ' || r.name || ' - ' || r.department || ' - ' || r.salary);

END LOOP;

DBMS\_OUTPUT.PUT\_LINE(CHR(10) || 'Accounts:');

FOR r IN (SELECT \* FROM accounts) LOOP

DBMS\_OUTPUT.PUT\_LINE(r.account\_id || ' - ' || r.customer\_name || ' - Balance: ' || r.balance);

END LOOP;

END;

/