**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.

**Create Table Accounts:**

CREATE TABLE ACCOUNTS (

ACCOUNTID NUMBER PRIMARY KEY,

CUSTOMERID NUMBER,

ACCOUNTTYPE VARCHAR2(20),

BALANCE NUMBER,

LASTMODIFIED DATE,

FOREIGN KEY ( CUSTOMERID )

REFERENCES CUSTOMERS ( CUSTOMERID )

);

**Create Table Transactions:**

CREATE TABLE TRANSACTIONS (

TRANSACTIONID NUMBER PRIMARY KEY,

ACCOUNTID NUMBER,

TRANSACTIONDATE DATE,

AMOUNT NUMBER,

TRANSACTIONTYPE VARCHAR2(10),

FOREIGN KEY ( ACCOUNTID )

REFERENCES ACCOUNTS ( ACCOUNTID )

);

**Create Table Employees:**

CREATE TABLE EMPLOYEES (

EMPLOYEEID NUMBER PRIMARY KEY,

NAME VARCHAR2(100),

POSITION VARCHAR2(50),

SALARY NUMBER,

DEPARTMENT VARCHAR2(50),

HIREDATE DATE

);

**Insert val1 into Accounts:**

INSERT INTO ACCOUNTS (ACCOUNTID, CUSTOMERID, ACCOUNTTYPE, BALANCE, LASTMODIFIED)

VALUES (1, 1, 'Savings', 1000, SYSDATE);

**Insert val2 into Accounts:**

INSERT INTO ACCOUNTS (ACCOUNTID, CUSTOMERID, ACCOUNTTYPE, BALANCE, LASTMODIFIED)

VALUES (2, 2, 'Checking', 1500, SYSDATE);

**Insert val1 into Transactions:**

INSERT INTO TRANSACTIONS (TRANSACTIONID, ACCOUNTID, TRANSACTIONDATE, AMOUNT, TRANSACTIONTYPE)

VALUES (1, 1, SYSDATE, 200, 'Deposit');

**Insert val2 into Transactions:**

INSERT INTO TRANSACTIONS (TRANSACTIONID, ACCOUNTID, TRANSACTIONDATE, AMOUNT, TRANSACTIONTYPE)

VALUES (2, 2, SYSDATE, 300, 'Withdrawal');

**Insert val1 into Employees:**

INSERT INTO EMPLOYEES (EMPLOYEEID, NAME, POSITION, SALARY, DEPARTMENT, HIREDATE)

VALUES (1, 'Alice Johnson', 'Manager', 70000, 'HR', TO\_DATE('2015-06-15', 'YYYY-MM-DD'));

**Insert val2 into Employees:**

INSERT INTO EMPLOYEES (EMPLOYEEID, NAME, POSITION, SALARY, DEPARTMENT, HIREDATE)

VALUES (2, 'Bob Brown', 'Developer', 60000, 'IT', TO\_DATE('2017-03-20', 'YYYY-MM-DD'));

**Scenerio 1 - CREATE OR REPLACE PROCEDURE :**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE accounts

SET balance = balance \* 1.01,

lastmodified = SYSDATE

WHERE UPPER(accounttype) = 'SAVINGS';

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('1% interest successfully added to all savings accounts.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Something went wrong: ' || SQLERRM);

END;

/

**Scenerio – 2:Update Employee Bonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept IN employees.department%TYPE,

p\_bonus\_percent IN NUMBER

) IS

BEGIN

UPDATE employees

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

hiredate = SYSDATE

WHERE department = p\_dept;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus of ' || p\_bonus\_percent || '% successfully applied to employees in department "' || p\_dept || '".');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error while applying bonus to department "' || p\_dept || '": ' || SQLERRM);

END;

/

**Scenerio – 3:Transfer Funds:**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from IN accounts.accountid%TYPE,

p\_to IN accounts.accountid%TYPE,

p\_amount IN NUMBER

) IS

v\_balance\_from accounts.balance%TYPE;

BEGIN

SELECT balance INTO v\_balance\_from

FROM accounts

WHERE accountid = p\_from

FOR UPDATE;

IF v\_balance\_from < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Transfer failed: Insufficient balance in account ' || p\_from || '.');

END IF;

UPDATE accounts

SET balance = balance - p\_amount,

lastmodified = SYSDATE

WHERE accountid = p\_from;

UPDATE accounts

SET balance = balance + p\_amount,

lastmodified = SYSDATE

WHERE accountid = p\_to;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Successfully transferred ₹' || p\_amount || ' from Account ' || p\_from || ' to Account ' || p\_to || '.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END;

/