**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.

**CREATION OF TABLES**

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

customer\_id NUMBER,

balance NUMBER(10, 2),

account\_type VARCHAR2(20)

);

CREATE TABLE employees (

emp\_id NUMBER PRIMARY KEY,

name VARCHAR2(50),

department VARCHAR2(30),

salary NUMBER(10, 2)

);

**INSERTION OF DATA**

INSERT INTO accounts VALUES (1, 101, 1000.00, 'Savings');

INSERT INTO accounts VALUES (2, 102, 2000.00, 'Current');

INSERT INTO accounts VALUES (3, 103, 3000.00, 'Savings');

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

INSERT INTO employees VALUES (2, 'Bob', 'Finance', 45000);

INSERT INTO employees VALUES (3, 'Charlie', 'HR', 42000);

**SCENARIO 1**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE accounts

SET balance = balance + (balance \* 0.01)

WHERE account\_type = 'Savings';

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

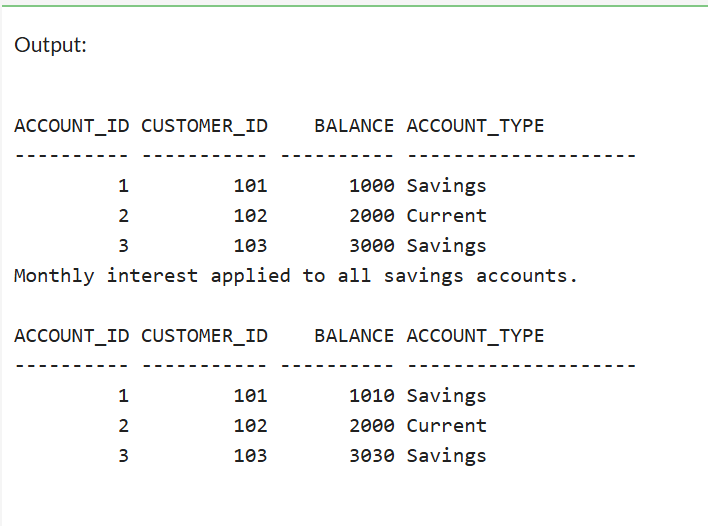
COMMIT;

END;

/

SELECT \* FROM accounts;

EXEC ProcessMonthlyInterest;



**SCENARIO 2**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

dept\_name IN VARCHAR2,

bonus\_pct IN NUMBER

) IS

BEGIN

UPDATE employees

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

WHERE department = dept\_name;

DBMS\_OUTPUT.PUT\_LINE(bonus\_pct || '% bonus applied to ' || dept\_name || ' department.');

COMMIT;

END;

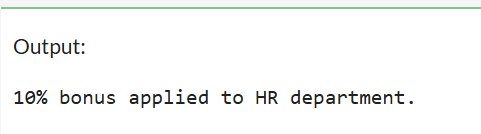
/

BEGIN

UpdateEmployeeBonus('HR', 10);

END;

/



**SCENARIO 3**

CREATE OR REPLACE PROCEDURE TransferFunds (

from\_account\_id IN NUMBER,

to\_account\_id IN NUMBER,

amount IN NUMBER

) IS

from\_balance NUMBER;

BEGIN

SELECT balance INTO from\_balance

FROM accounts

WHERE account\_id = from\_account\_id

FOR UPDATE;

IF from\_balance < amount THEN

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

END IF;

UPDATE accounts

SET balance = balance - amount

WHERE account\_id = from\_account\_id;

UPDATE accounts

SET balance = balance + amount

WHERE account\_id = to\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Transferred Rs ' || amount || ' from account ' || from\_account\_id || ' to account ' || to\_account\_id);

COMMIT;

END;

/

EXEC TransferFunds(1, 2, 500);

