**Scenario 1: Process Monthly Interest for All Savings Accounts**

This stored procedure ProcessMonthlyInterest will calculate and update the balance of all savings accounts by applying an interest rate of 1% to the current balance.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

-- Update the balance of all savings accounts by applying a 1% interest

UPDATE accounts

SET balance = balance \* 1.01

WHERE account\_type = 'SAVINGS';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_logs(message, log\_date)

VALUES (SQLERRM, SYSDATE);

RAISE;

END ProcessMonthlyInterest;

/

**Scenario 2: Implement a Bonus Scheme for Employees**

This stored procedure UpdateEmployeeBonus will update the salary of employees in a given department by adding a bonus percentage passed as a parameter.

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_department\_id IN NUMBER,

p\_bonus\_percentage IN NUMBER

) IS

BEGIN

-- Update the salary of employees in the given department by adding the bonus percentage

UPDATE employees

SET salary = salary \* (1 + p\_bonus\_percentage / 100)

WHERE department\_id = p\_department\_id;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_logs(message, log\_date)

VALUES (SQLERRM, SYSDATE);

RAISE;

END UpdateEmployeeBonus;

/

**Scenario 3: Transfer Funds Between Accounts**

This stored procedure TransferFunds will transfer a specified amount from one account to another, checking that the source account has sufficient balance before making the transfer.

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

insufficient\_funds EXCEPTION;

PRAGMA EXCEPTION\_INIT(insufficient\_funds, -20001);

BEGIN

-- Check if the source account has sufficient funds

DECLARE

v\_balance NUMBER;

BEGIN

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

IF v\_balance < p\_amount THEN

RAISE insufficient\_funds;

END IF;

-- Perform the transfer

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account\_id;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account\_id;

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

ROLLBACK;

INSERT INTO error\_logs(message, log\_date)

VALUES ('Insufficient funds in account ' || p\_from\_account\_id, SYSDATE);

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

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_logs(message, log\_date)

VALUES (SQLERRM, SYSDATE);

RAISE;

END;

END TransferFunds;

/

**Summary**

* **Scenario 1**: ProcessMonthlyInterest updates the balance of all savings accounts by applying a 1% interest rate.
* **Scenario 2**: UpdateEmployeeBonus updates the salary of employees in a given department by adding a bonus percentage.
* **Scenario 3**: TransferFunds ensures sufficient balance before transferring funds from one account to another and logs errors appropriately.